Face Recognition via Ensemble SIFT Matching of Uncorrelated Hyperspectral Bands and Spectral PCTs

THESIS

Mohd Fairul Mohd-Zaid

AFIT-OR-MS-ENS-11-16

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the United States Government.

AFIT-OR-MS-ENS-11-16

Face Recognition via Ensemble SIFT Matching of Uncorrelated Hyperspectral Bands and Spectral PCTs

THESIS

Presented to the Faculty

Department of Operational Sciences

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Operations Research

Mohd Fairul Mohd-Zaid, BS

June 2011

AFIT-OR-MS-ENS-11-16

Face Recognition via Ensemble SIFT Matching of Uncorrelated Hyperspectral Bands and Spectral PCTs

Mohd Fairul Mohd-Zaid, BS

Approved:

_______//Signed// _____________________ 16 June 2011__
Dr. Kenneth W. Bauer (Chairman) date

________//Signed______________________ 16 June 2011__
Maj. Mark A. Friend (Member) date

AFIT-OR-MS-ENS-11-16

Abstract

Face recognition through hyperspectral images is a concept which is still in its infancy. Although the conventional method of face recognition using Red-Green-Blue (RGB) or grayscale images has been advanced over the last twenty years, these methods are still shown to have weak performance whenever there are variations or changes in lighting, pose, or temporal aspect of the subjects. A hyperspectral representation of an image captures more information that is available within a scene than a RGB image therefore it is beneficial to study the performance of face recognition using a hyperspectral representation of the subjects' faces.

We studied the results of a variety of methods for performing face recognition using the Scale Invariant Transformation Feature (SIFT) algorithm as a matching function on uncorrelated spectral bands, principal component representation of the spectral bands, and the ensemble decision of the two. We conclude that there is no dominating method in the scope of our research; however, we do obtain three methods that outperform the results obtained from a previous study which only considered a SIFT application on a single hyperspectral band, and our method performs very well under temporal variation.

AFIT-OR-MS-ENS-11-16

To my parents,
for giving me the opportunity to dream.

Acknowledgments

I would like to first thank Dr. Kenneth Bauer for his continuous guidance and support over the course of my research despite the occasional lack of results, on my behalf, presented during some of our many meetings. I would also like to thank him for giving myself the creativity freedom in performing my research. Secondly, I would like to thank Major Mark Friend for his knowledgeable inputs and insights, especially, in the area of ensemble classification which further improved my research.

I would also like to thank Lt Col Dave Ryer for his many assistances and insights with respect to SIFT and the CMU dataset. Last but not least I would like to give a special mention to the members of the Sensor Fusion Lab for their technical help and ideas and also to my classmates and friends who have made my time at AFIT more enjoyable.

Fairul Mohd-Zaid

Table of Contents

List of Figures

List of Tables

FACE RECOGNITION VIA ENSEMBLE SIFT MATCHING OF UNCORRELATED HYPERSPECTRAL BANDS AND SPECTRAL PCTS

1. Introduction

1.1. Background

Face recognition is not a fairly new area of study, but facial recognition using hyperspectral images is a concept which is still in its infancy. Principal Component Analysis (PCA) was first used in the context of face recognition some 20 years ago whereas hyperspectral face recognition first surfaced in the early 2000s (Pan Z. , Healey, Prasad, & Tromberg, 2003). One of the first methods used with respect to face recognition and hyperspectral images involved comparing the hyperspectral signatures from different components, obtained either manually or by using the K-means clustering, as proposed by Elbakary (2007), of the face between the probe and the target images. The state-of-the-art method for face recognition algorithm evaluation is the FERET method of evaluation which was developed in the late 90s where instead of measuring performance using a ROC curve, accuracy is measured against the rank at which the true image in the gallery is matched with the probe. Initially designed for object recognition, the Scale-Invariant Feature Transform (SIFT) extracts descriptors from the ‘scale space’ of an image and is quite robust to changes in illumination, 3D rotation, and scale of an object and it has been shown that it performs well even with RGB images (Lowe, 2004).

1.2. Problem Statement

One of the lingering problems within face recognition is performance degradation when illumination variation or temporal changes are present as stated by Phillips et al. (1997), Pan et al. (2003), and Luo et al. (2007). Although it has been shown by Luo et al. (2007) that an implementation of SIFT does not work well under illumination and temporal variation, no literature has been found that studies the performance of SIFT with hyperspectral images. Therefore it is advantageous for us to investigate the possible benefits of such method.

1.3. Approaches

The goal of this research is to perform face recognition using all of the tools listed above and compare the result with the results obtained from previous researches. Autocorrelation between the spectral bands of the images is first studied to reduce the dimensionality of the image by only retaining bands that are uncorrelated to each other. Descriptors are then extracted from each remaining bands using SIFT and face recognition is then performed by matching the descriptors of a target in the gallery with the descriptors of the probe for each uncorrelated bands. Each band-matching yield a number of possible descriptor-matches, and the descriptor-matches from each band are fused to give an ensemble matching score for the target and probe. There are six parameters in the algorithm that alter the matching accuracy and the runtime of the algorithm, therefore a design of experiment is performed to determine if any of the parameters are insignificant and to determine the optimal setting for the algorithm.

Intuitively, applying SIFT to the first few principal components of the hyperspectral image should give a similar result since the principal components are orthogonal and each provides information that the others do not (i.e. they are uncorrelated). The next step is to find out if similar or better results can be obtained by SIFT-PCT with a fewer number of principal components than the number of uncorrelated bands since this would require less iterations of SIFT and is computationally quicker. The accuracy of the final algorithm will then be tested using the FERET method and compared with other available results.

1.4. Research Objectives

The objective of this research is to investigate the benefits of applying SIFT to hyperspectral images in the context of face recognition. In particular, performance with respect to temporal variation is of significant concern provided that high baseline performance can be met. We will then compare the results that we obtain to other available research using the same dataset used in our research.

2. Literature Review

2.1. Hyperspectral Imaging

A very basic form of a digital image is a black and white image from a camera. A black and white digital image displays the relative intensity level light in the pixel. A color image can be thought of three monochromatic images merged together with different wavelength bands being used to represent what our eyes see as red, green, and blue. The camera essentially collects three images. When a hyperspectral image is created the scene is recorded with up to 250 wavelength bands. These bands normally extend from the visible region (0.4-0.7 µm) into the near infrared region (0.7-1.1 µm) and shortwave infrared region (1.1-3.0 µm) of the electromagnetic spectrum (0.7-2.5 µm) (Landgrebe, 2003). Some hyperspectral sensors are configured to collect midwave (3.0-5.0 µm) and longwave infrared (5.0-15 µm) (Eismann, 2010). Figure 2.1-1 shows the segment of the electromagnetic spectrum used for hyperspectral imaging. The increased number of collected wavelengths allow for the comparison of materials that would not be distinguishable with a lower number of collected wavelengths (Shaw & Manolakis, 2002).

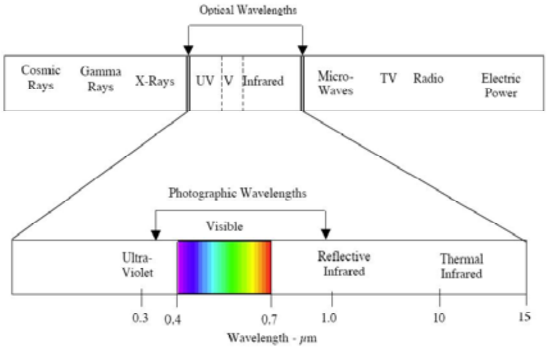


Figure 2.1-1: The Electromagnetic Spectrum (Landgrebe, 2003, p. 14).

2.2. Facial Recognition

Most current face recognition systems discriminate faces using the distinctive geometric or biometric features of each individual face. However, such a method of facial recognition shows degradation in performance when variability in illumination, facial orientation, or temporal changes are present as shown by Phillips et al. (1997), Zhao et al. (2003), and Luo et al. (2007). Spectral properties of the human face have been shown to be a good tool for facial recognition. This leads to method that performs surprisingly well for front-view face images of variant facial expressions, where only a few local reflectance spectra were used for discrimination (Pan, Healey, & Tromberg, 2005, pp. 144-145). In this section, we will expose the reader to some of the work that has already been done in this area which includes using eigenfaces for recognition by Turk and Pentland (1991), the standardized Face Recognition Technology (FERET) method of evaluation for face recognition algorithms by Phillips et al. (1997), and recent works in performing face recognition using hyperspectral images by Pan et al. (2003, 2005).

2.2.1. Eigenfaces

Turk and Pentland suggest that developing a computational model of face recognition is quite difficult, because faces are complex, multidimensional, and meaningful visual stimuli. Therefore, a pre-attentive pattern recognition capability that does not depend upon having full three-dimensional models is therefore useful (Turk & Pentland, 1991, p. 71). Using Principal Component Analysis (PCA), the eigenvectors of the covariance matrix of the dataset (set of images) can be found by treating each picture as a vector. Each picture can then be represented as a linear combination of the best eigenvectors or "eigenfaces" which has the largest eigenvalues that account for the most variance. Motivated by Sirovich and Kirby (1987) and Kirby and Sirovich (1990), they proposed that weights of the linear combination of each face can be obtained by projecting the face onto each eigenface. Comparison can then be made between the weights of a target face and weights of known individuals by transforming an n by m pixel image into a single array, Γ_n, and letting

$$\mathbf{\Psi} = \frac{1}{M}\sum\nolimits_{n=1}^{M} \mathbf{\Gamma}_n$$

$$\mathbf{\Phi}_i = \mathbf{\Gamma}_i - \mathbf{\Psi} ,$$

where

$\mathbf{\Gamma}_i$ is a set of training images for $i = 1, \dots, M$

$\mathbf{\Psi}$ is the average face

$\mathbf{\Phi}_i$ is the mean $-$ average face.

The covariance matrix of the dataset can then be calculated by as follows

$$C = \frac{1}{M}\sum_{n=1}^{M} \mathbf{\Phi}_n \mathbf{\Phi}_n^T = AA^T$$

Where

$A = [\mathbf{\Phi}_1\ \mathbf{\Phi}_2 \dots \mathbf{\Phi}_M]$ is a matrix of mean $-$ adjusted face of each subject

Consider then the eigenvectors $\boldsymbol{v}_i\ of\ A^T A$ such that

$$A^T A \boldsymbol{v}_i = \mu_i \boldsymbol{v}_i$$

which if premultiplied by A is equal to

$$AA^T A\boldsymbol{v}_i = \mu_i A\boldsymbol{v}_i$$

So $A\boldsymbol{v}_i$ are eigenvectors of the covariance matrix $C = AA^T$. So if we can find the eigenvectors of $A^T A$, we can find the eigenvectors of C by premultiplying them by A which is computationally simpler since $A^T A$ is smaller than C. The eigenfaces $\boldsymbol{u}_k$ of C can then be computed by

$$\boldsymbol{u}_k = \sum_{l=1}^{M} \boldsymbol{v}_{kl}\boldsymbol{\Phi}_l,\ \ k = 1, \dots, M$$

A new face image $\boldsymbol{\Gamma}$ can then be transformed into its eigenface components (ω_k) by the following operation

$$\omega_k = \boldsymbol{u}_k^T(\boldsymbol{\Gamma} - \boldsymbol{\Psi}),\ \ k = 1 \dots M',$$

where M' is the number of best eigenface and $M' < M$. The weight vector for a face can then be defined as

$$\Omega^{\mathrm{T}} = [\omega_1, \omega_2, \dots, \omega_{M'}].$$

We can then use the Eucledian distance to perform face recognition by comparing a target's weight vector with that of a specific face class (a known face) and assigning the target to a individual *k* if the distance is below a certain threshold; if not then the target can be assigned as "unknown". We can also detect whether or not an image contains a face at all by calculating the distance $\boldsymbol{\epsilon}$ of the mean-adjusted target image to that of the its projection into the "face space" which can be defined as

$$\boldsymbol{\epsilon}^2 = ||\boldsymbol{\Phi} - \boldsymbol{\Phi}_{\mathrm{f}}||^2,$$

where

$$\boldsymbol{\Phi}_{\mathrm{f}} = \sum\nolimits_{\mathrm{i}=1}^{\mathrm{M}'} \omega_{\mathrm{i}}\mathbf{u}_{\mathrm{i}}.$$

Therefore there are four possibilities for a target image which are: (1) Near face space and near face class (recognized), (2) Near face space but not near known face class (unknown), (3) Not near face space but near face class (false positive), and (4) Not near face space and not near face class (not a face). Figure 2.2-1 helps illustrate the four possibilities for a particular target.

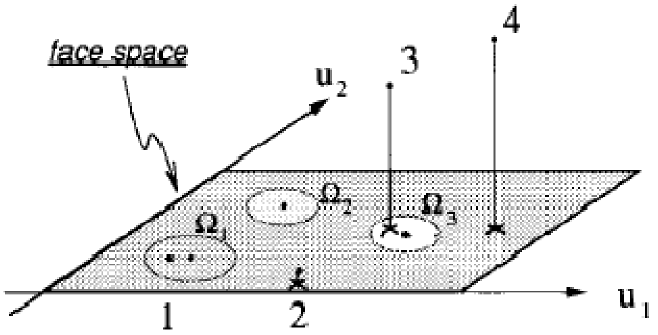


Figure 2.2-1: A simplified version of face space to illustrate the four results of projecting an image into face space. In this case, there are two eigenfaces (u_1 and u_2) and three known individuals (Ω_1, Ω_2, and Ω_3) (Turk & Pentland, 1991, p. 589).

Turk and Pentland (1991) showed that change in lighting causes relatively few errors in comparison to changing head size and this is because neighborhood pixel correlation remains high under lighting condition but is low when varying the head size. The aim of their research was to develop a computational model of face recognition which is fast, reasonably simple, and accurate in constrained environments such as an office or a household and to automatically learn and recogne new faces is practical within this framework (Turk & Pentland, 1991). They also note that a noisy image or partially occluded face should cause recognition performance to degrade slightly, but not significantly since the system essentially implements an autoassociative memory for the known faces. Figure 2.2-2 shows an example where an occluded face image can be reconstructed using eigenfaces to closely resemble the true image.

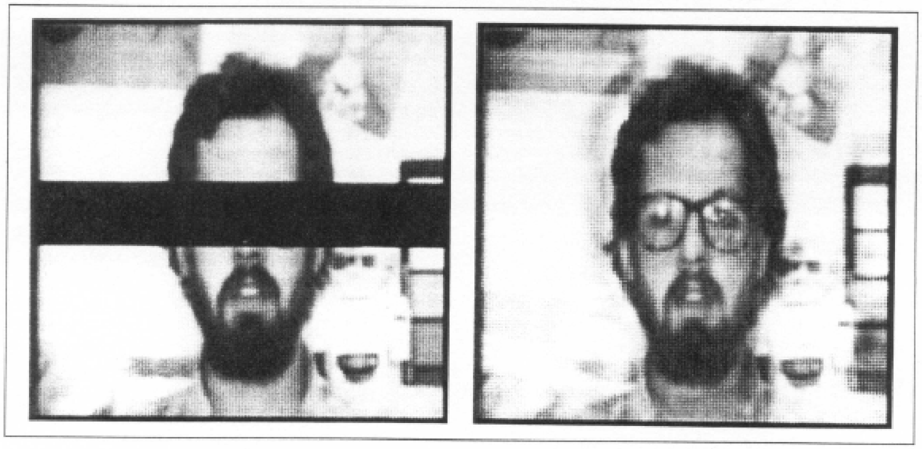


Figure 2.2-2: (a) Partially occluded face image and (b) its reconstruction using eigenfaces (Turk & Pentland, 1991, p. 85).

The eigenface approach to face recognition was motivated by information theory, leading to the idea of basing face recognition on a small set of image features that best approximate the set of known face images, without requiring that they correspond to our intuitive notions of facial parts and features. Although it is not an elegant solution to the general object recognition problem due to the intensiveness of the computation, the eigenface approach does provide a practical solution that is well fitted to the problem of face recognition. It is fast, relatively simple, and has been shown to work well in a somewhat constrained environment.

2.2.2. The FERET Evaluation Method

Two of the most critical requirements in producing reliable face-recognition systems are a large database of facial images and a testing procedure to evaluate systems. The FERET program has addressed both issues through establishing the FERET tests which forms a general evaluation tool designed to measure performance of laboratory algorithms on the FERET database. The first FERET tests took place in August 1994 and March 1995 whereas the FERET database collection began in September 1993 along with the FERET program. The program and database were created to provide a common database of images and method of evaluation for both developing and testing face recognition algorithms.

As of July 1996, the FERET Image Database contains 14,126 total images from 1199 individuals separated into 1564 sets. Each set consists of five to eleven images of a single subject taken under specific pose and illumination variations. 503 of these sets are available for development purposes and the remaining 1061 are government sequestered (Phillips, Moon, Rauss, & Rizvi, 1997). Below is a list of definitions that will be used for the remaining of this section:

Gallery - set of known individuals.

Probe - an unknown image presented to the algorithm.

Duplicate - an image of a person whose corresponding gallery image was taken on a different date some of which are more than 2 years apart.

Images – **FA** (front), **FB** (front w/different expression), **FC** (front w/different lighting).

Also, there are two methods of evaluation used in evaluating the various algorithms presented:

Closed universe (every probe is in gallery) - score is R_k/P, where P is the number of probes to be scored and R_k is the number of correctly matched probes in the subset of P that are in the top k rank.

Open universe (some probes not in gallery) - results are reported on a ROC that shows trade-off between false alarm rate and the probability of correct identification.

There were eight algorithms presented for the purpose of evaluation, one of which is the Army Research Lab (ARL) implementation of the eigenface method. Other algorithms include those by Excalibur Corporation, Massachusetts Institute of Technology (MIT), Rutgers University, University of Maryland, and Michigan State University. There were two versions of the algorithm presented by MIT, respectively dated March 1995 and September 1996. The latest algorithm outperformed all of the other algorithms that were evaluated. The results from the test suggests that images from the same set (FA versus FB) are the least difficult to recognize (93% at Rank 10) and images taken a year or more apart (duplicates) are the most difficult (45% at Rank 10) (Phillips, Moon, Rauss, & Rizvi, 1997, p. 143). For the open universe evaluation, ARL implementation of the eigenface method does not perform as well, but performs above average in closed universe with duplicate probes. Algorithm performance is dependent on the gallery and probe sets. Figure 2.2-3 and 2.2-4 are shown below to compare the results and Table 2.2-1 and 2.2-2 lists the ranking of the evaluated algorithms for each tests.

Some of the conclusions of this research are that there are still significant challenges in recognizing faces from duplicate images, handling variations due to illumination, and in understanding how changing the gallery affects algorithm performance. Another test was conducted in 1997 and paper published by Phillips et al.. (2000) with similar results.

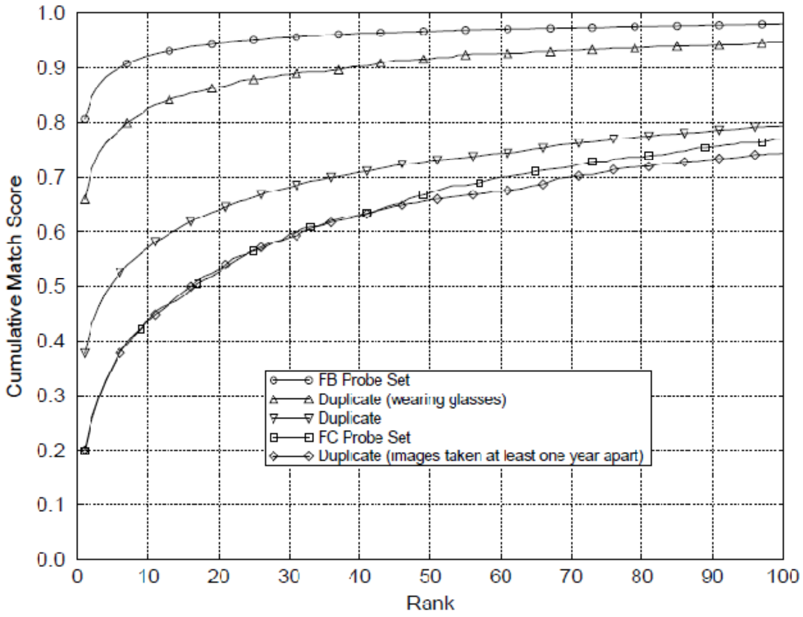


Figure 2.2-3: Comparison of performance against different categories of probes (Phillips, Moon, Rauss, & Rizvi, 1997, p. 141)

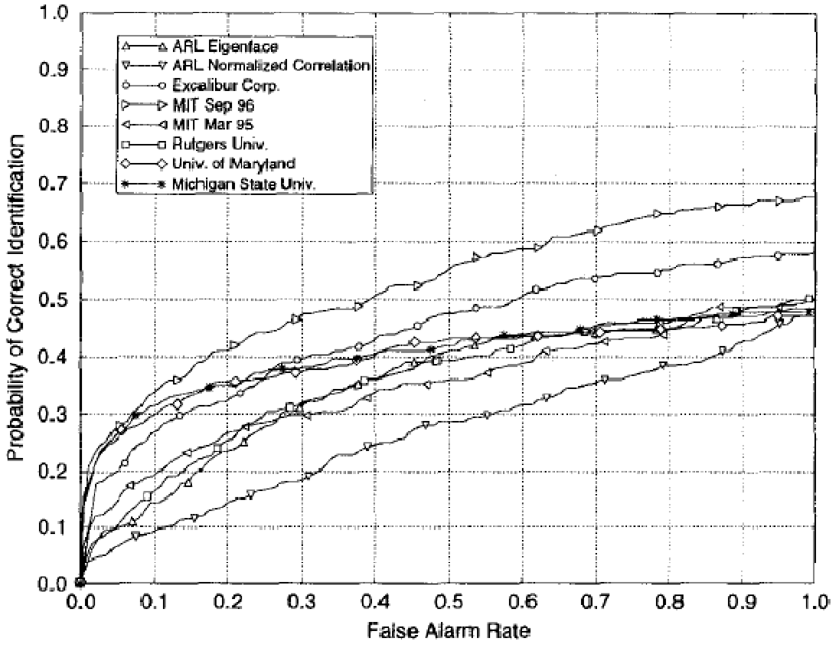
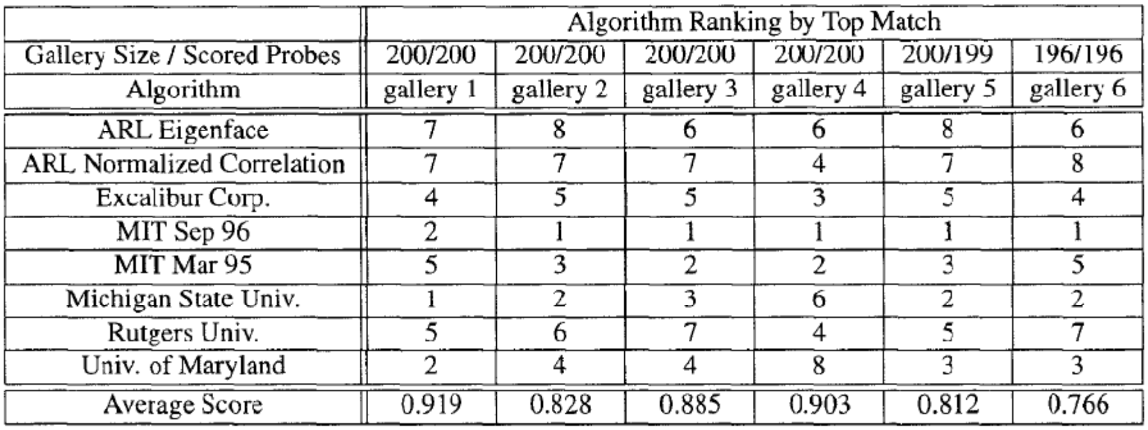


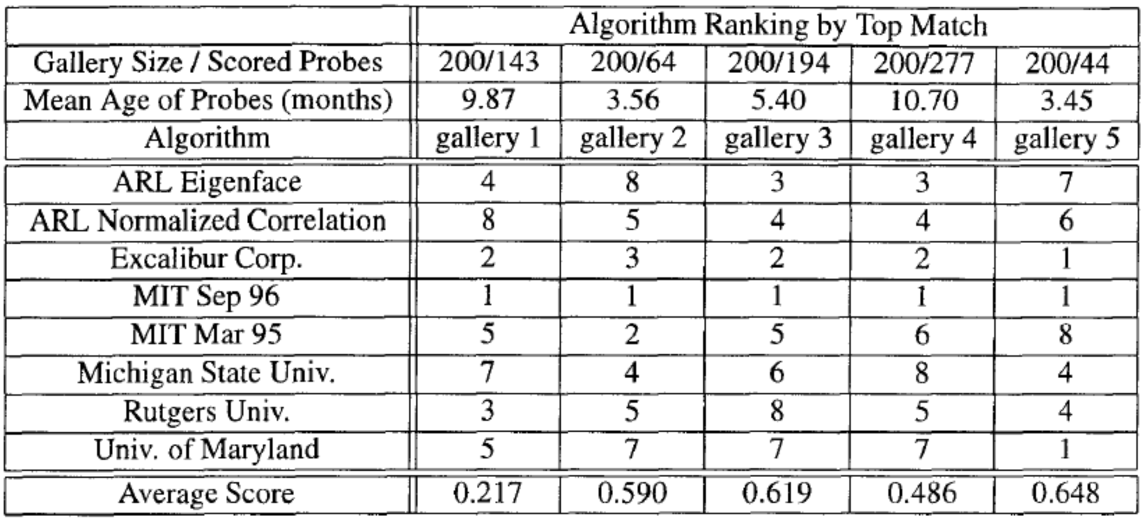
Figure 2.2-4: Performance of algorithms on false-alarm test. (Gallery size = 100, number of probes = 2927, number of duplicate probes belonging to gallery = 309.) (Phillips, Moon, Rauss, & Rizvi, 1997, p. 141)

Table 2.2-1: Variations in performance on 6 different galleries on FB probes. Images in each gallery do not overlap (Phillips, Moon, Rauss, & Rizvi, 1997, p. 142).



	Algorithm Ranking by Top Match					
Gallery Size / Scored Probes	200/200	200/200	200/200	200/200	200/199	196/196
Algorithm	gallery 1	gallery 2	gallery 3	gallery 4	gallery 5	gallery 6
ARL Eigenface	7	8	6	6	8	6
ARL Normalized Correlation	7	7	7	4	7	8
Excalibur Corp.	4	5	5	3	5	4
MIT Sep 96	2	1	1	1	1	1
MIT Mar 95	5	3	2	2	3	5
Michigan State Univ.	1	2	3	6	2	2
Rutgers Univ.	5	6	7	4	5	7
Univ. of Maryland	2	4	4	8	3	3
Average Score	0.919	0.828	0.885	0.903	0.812	0.766

Table 2.2-2: Variations in performance on 5 different galleries on duplicate probes. Images in each of the gallery do not overlap (Phillips, Moon, Rauss, & Rizvi, 1997, p. 143).



	Algorithm Ranking by Top Match				
Gallery Size / Scored Probes	200/143	200/64	200/194	200/277	200/44
Mean Age of Probes (months)	9.87	3.56	5.40	10.70	3.45
Algorithm	gallery 1	gallery 2	gallery 3	gallery 4	gallery 5
ARL Eigenface	4	8	3	3	7
ARL Normalized Correlation	8	5	4	4	6
Excalibur Corp.	2	3	2	2	1
MIT Sep 96	1	1	1	1	1
MIT Mar 95	5	2	5	6	8
Michigan State Univ.	7	4	6	8	4
Rutgers Univ.	3	5	8	5	4
Univ. of Maryland	5	7	7	7	1
Average Score	0.217	0.590	0.619	0.486	0.648

2.2.3. Face Recognition in Hyperspectral Imaging

Spectral properties of human tissue vary significantly from person to person and this uniqueness can be used to an advantage in face recognition. Pan (2003) presented a method of matching tissue based on hyperspectral signatures extracted from different areas of the face. The experiments conducted by Pan consider a database of calibrated near-infrared hyperspectral images for 200 subjects where the image collection is similar to FERET consisting sets of seven images per subject and 20 duplicates. There are 31 bands for each image sampled at 10 nm over 700 nm -1000 nm with a pixel resolution of 468x494. Figure 2.2-5 and 2.2-6 shows an example of images associated with each subject.

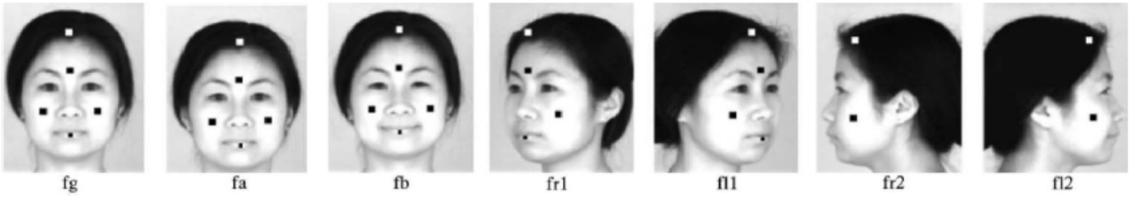


Figure 2.2-5: Examples of images with different expressions and rotations (Pan Z. , Healey, Prasad, & Tromberg, 2003).

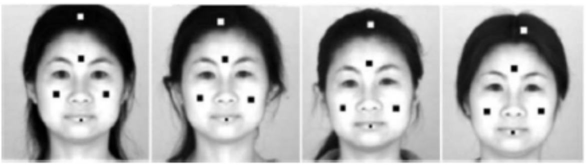


Figure 2.2-6: Examples of duplicates (taken at different times) (Pan Z. , Healey, Prasad, & Tromberg, 2003)

It was noted that the raw images had to be converted to spectral reflectance images with pixel reflectance, $R(x, y, \lambda_k)$, due to unknown gains from filter transmission and charged-coupled device (CCD) response and unknown offsets from stray lights. The conversion method is excluded from this paper and the reader is referred to the original paper for further reading (Pan Z. , Healey, Prasad, & Tromberg, 2003). A face recognition algorithm based on the spectral comparison of combinations of tissue types was then established and applied to the images. Pan et al. defined

$$\boldsymbol{R}_t = \left(R_t(\lambda_1), R_t(\lambda_2), \ldots, R_t(\lambda_B)\right)^T$$

to be the spectral reflectance vector for each tissue type t where

$$R_t(\lambda_k) = \frac{1}{N}\sum_{x,y} R(x, y, \lambda_k)$$

$$t \in \{\text{hair}, \text{forehead}, \text{left cheek}\,, \text{right cheek}, \text{lip}\}$$

and B is the total number of bands. If we obtain the normalized spectral reflectance vector $\overline{\boldsymbol{R}}_t$ for tissue type t by letting

$$\overline{\boldsymbol{R}}_t = \frac{\boldsymbol{R}_t}{\|\boldsymbol{R}_t\|},$$

we can then use the Mahalanobis distance to measure the distance between target image and test image for each tissue type by

$$D_t^{'}(i,j) = (\overline{\boldsymbol{R}}_t(i) - \overline{\boldsymbol{R}}_t(j))^T\ \Sigma_{\mathrm{t}}^{-1}\ (\overline{\boldsymbol{R}}_t(i) - \overline{\boldsymbol{R}}_t(j))$$

where Σ_{t} is the covariance matrix of $\overline{\boldsymbol{R}}_t$ and (i,j) is a pairing of target and test images. $D_t^{'}(i,j)$ is dependent on the locations of the neighborhood squares used to compute $\overline{\boldsymbol{R}}_t(i)$ since the tissue spectral reflectance can have spatial variability meaning that the same pixel coordinate from two different images of a single subject may not be the same point on the subject itself. To address this issue, a modified form of $D_t^{'}(i,j)$ is introduced by the Pan as follows

$$D_t(i,j) = \min_{k\in[1,M],l\in[1,M]} \left[\left(\overline{\boldsymbol{R}}_t^{(k)}(i) - \overline{\boldsymbol{R}}_t^{(l)}(j)\right)^T\ \Sigma_{\mathrm{t}}^{-1}\ \left(\overline{\boldsymbol{R}}_t^{(k)}(i) - \overline{\boldsymbol{R}}_t^{(l)}(j)\right)\right]$$

where

$$M = number\ of\ adjacent\ squares.$$

To utilize all visible tissue types, a total distance function is introduced that includes the distances for all visible tissue types defined as

$$D(i,j) = \omega_f D_f(i,j) + \omega_{lc} D_{lc}(i,j) + \omega_{rc} D_{rc}(i,j) + \omega_h D_h(i,j) + \omega_l D_l(i,j)$$

where

$$\omega_t = \begin{cases} 1, & \text{if } t \text{ is visible} \\ 0, & \text{otherwise} \end{cases}.$$

The FERET evaluation method for scoring is then used and results between various poses and type of images. Only frontal view image of each subject was used in the gallery and the rest of the images were used as probes and only the closed universe evaluation was performed thus no acceptance threshold was used and the smallest distance is assigned as a match. The results from the experiment showed that the algorithm provides accurate recognition performance for expression changes and for images acquired over several week time intervals as shown in Figure 2.2-7 and 2.2-8 where different expression has slightly lower match score 89% versus same expression 94%, and duplicates acquired within one week (40 probes) has similar score to duplicates acquired over one week (58 probes) of 92% at rank 10. However, that is a relatively short period of time compared to FERET experiment.

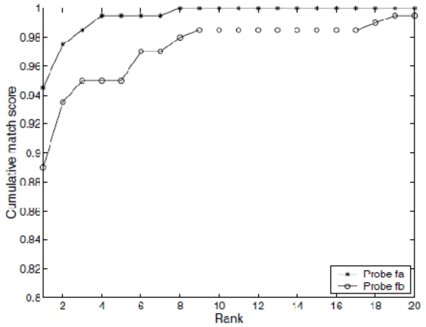


Figure 2.2-7: Probe comparison of fa and fb (Pan Z. , Healey, Prasad, & Tromberg, 2003).

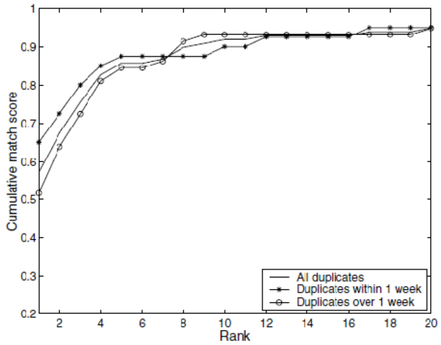


Figure 2.2-8: Peformance of duplicates (Pan Z. , Healey, Prasad, & Tromberg, 2003).

It was also shown that skin tissue type has better match score versus hair and lips however the combined score provided a jump in performance as shown in Figure 2.2-9. Figure 2.2-10 shows the result for view rotation where the front view orientation has better match score (92%) versus 45 degree (75%) and 90 degree (51%). Since the algorithm uses only local spectral information, it is expected that additional performance gains can be achieved by incorporating spatial information into the recognition process and previous work by Healey and Slater (1999) has shown that the high-dimensionality of hyperspectral data supports the use of subspace methods for illumination-invariant recognition.

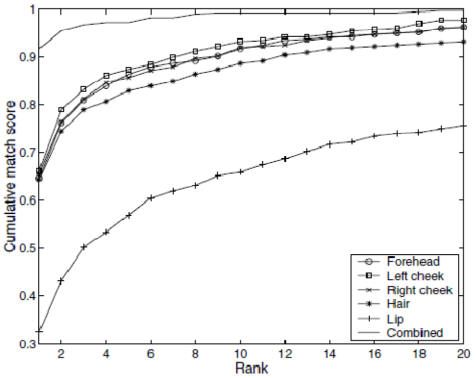


Figure 2.2-9: Peformance using both fa and fb (Pan Z. , Healey, Prasad, & Tromberg, 2003).

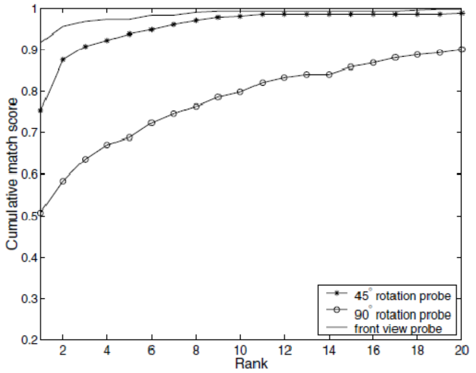


Figure 2.2-10: Identification performance of rotated faces (Pan Z. , Healey, Prasad, & Tromberg, 2003).

Pan et al. (2005) then proposed three other method of utilizing hyperspectral images for face recognition. Their previous work as demonstrated above has shown that spectral signatures are powerful discriminants for face recognition in hyperspectral images. Since no dataset of hyperspectral face images similar to the FERET dataset exists, they had to utilize the same hyperspectral face images used in their previous research where it is shown that the method of using local spectral properties provides excellent results with minimal computation. The FERET method of evaluation used in their latter research is the closed universe method which only looks at frontal view images with no duplicates. The three methods of face recognition for the follow up study as coined by Pan et al. are: Single-Band Images, Multiband Images, and Spectral Eigenfaces.

Single-Band Images

The CSU Face Identification Evaluation System created by Beveridge et al. (2004) provides a data transformation feature, algorithm selection (eigenfaces method was selected in the research conducted by Pan et al.) and method of scoring (FERET) was used. Gray scale images were extracted from each of the 31 bands. The CSU evaluation system transformed and normalized each image to 130x150 with uniform eye coordinates and applied an ellipse mask to void non-facial features. All 600 images were used to generate the set of eigenfaces. The number of eigenfaces used for recognition is based on number of eigenvalues that account for 90% of the total variance.

Suppose there are a total of 31 bands between spectral range of 700-1000 nm at a width of 10 nm per band. Given a hyperspectral image $\boldsymbol{u}$, let $u_{w,i}$ be the projection of the w^{th} band of $\boldsymbol{u}$ onto the i^{th} eigenface which can be obtained by

$$u_{w,i} = \boldsymbol{v}_{w,i}^{T}(\boldsymbol{u}_w - \boldsymbol{\Psi}_{\mathrm{w}})$$

where $\boldsymbol{v}_{w,i}$ is the i^{th} eigenface of all w^{th} band training images, $\boldsymbol{u}_w$ is the w^{th} band image of $\boldsymbol{u}$ and $\boldsymbol{\Psi}_{\mathrm{w}}$ is the average image of the w^{th} band of training images. Also, let $\sigma_{w,i}$ be the standard variation of projections from all w^{th} band images onto the i^{th} eigenface. If we define

$$\boldsymbol{u}_w = \left(u_{w,1}, u_{w,2}, \dots, u_{w,I}\right),$$

then we have the Mahalanobis projection of $\boldsymbol{u}_w$ as

$$\boldsymbol{m}_w = (m_{w,1}, m_{w,2}, \dots, m_{w,I})$$

where

$$m_{w,i} = \frac{u_{w,i}}{\sigma_{w,i}}.$$

The Mahalanobis Cosine distance between $\boldsymbol{u}$ and $\boldsymbol{v}$ at the w^{th} band is defined as

$$D_{\boldsymbol{u},\boldsymbol{v}}(w) = -\frac{\boldsymbol{m}_w \cdot \boldsymbol{n}_w}{|\boldsymbol{m}_w||\boldsymbol{n}_w|}$$

which is the negative of the cosine function between two vectors and has a range of [-1,1] and increases from -1 for identical matches to 1 for opposite matches.

The single-band eigenface method used spatial features exclusively (eigenface) and performed noticeably better than pure spectral method as shown in Figure 2.2-12. However spectral method in pose variant showed promising result from previous study. Also the computation task increases significantly for eigenfaces generation and projection.

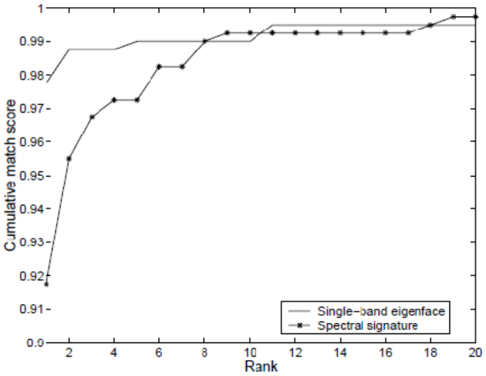


Figure 2.2-11: Cumulative match scores of spectral signature method and the best of single-band eigenface method (Pan, Healey, & Tromberg, 2005, p. 147)

Multiband Images

Pan et al. has shown that separately spatial and spectral features in hyperspectral images are good discriminants and an improvement could be attained by combining both. One way to accomplish this is by defining a new distance function as

$$\boldsymbol{D_{u,v}} = \sqrt{\sum_{w=1}^{W} (1 + D_{\boldsymbol{u,v}}(w))^2}$$

where W is a total of selected bands and the addition of 1 is to ensure a positive sum. We can also consider reducing the dimensionality of the hyperspectral image using Principal Component Transformation (PCT) by transforming a hyperspectral image

$$\boldsymbol{u} = (\boldsymbol{u}_1, \boldsymbol{u}_2, \dots, \boldsymbol{u}_W),$$

where $\boldsymbol{u}_W$ is the w^{th} band image of $\boldsymbol{u}$ as previously defined, to

$$\boldsymbol{u}' = (\boldsymbol{u}_1', \boldsymbol{u}_2', \dots, \boldsymbol{u}_W')$$

by letting

$$\boldsymbol{u}_i' = \sum\nolimits_j \epsilon_{ij} \boldsymbol{u}_j$$

where ϵ_{ij} is the ij^{th} element of the Principal Component of the set of images of $\boldsymbol{u}$. We can then denote $\boldsymbol{u}'$ as the 'Principal Bands' of $\boldsymbol{u}$ that accounts for the most spectral variations over the defined spectral bands. Figure 2.2-12 demonstrates the five principal band images of one subject obtained using PCT.

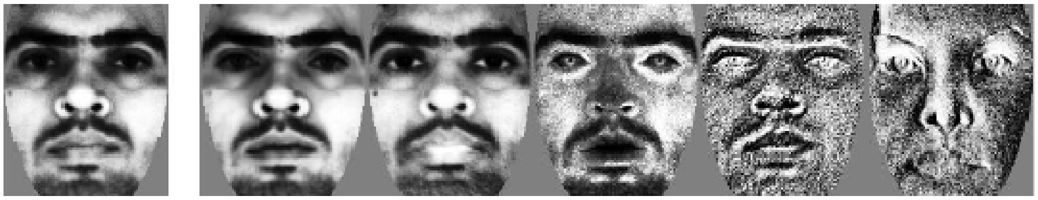


Figure 2.2-12: Images of one subject obtained using PCT of the hyperspectral bands (Pan, Healey, & Tromberg, 2005, p. 147)

The recognition rate was further improved with the multiband methods as shown in Figure 2.2-13 and 14 but it demands more computation power. The best performance was achieved with the highest computation complexity when the first three principal bands were used together. A drawback to this method is that reducing a hyperspectral image with 31 bands to 3 principal bands using principal component analysis is computationally expensive.

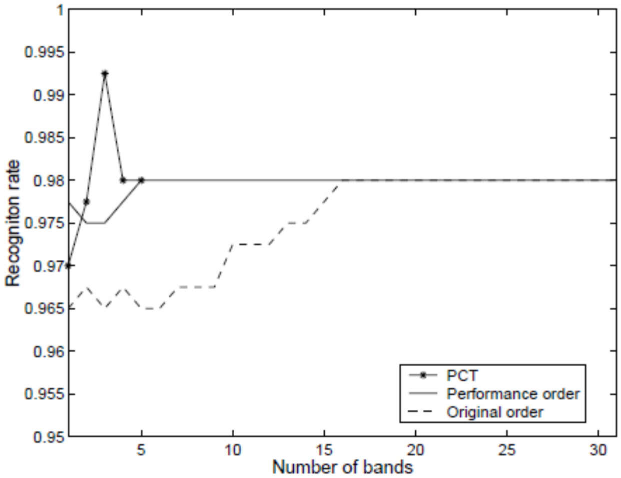


Figure 2.2-13: Performance based on number of bands included (Pan, Healey, & Tromberg, 2005, p. 148).

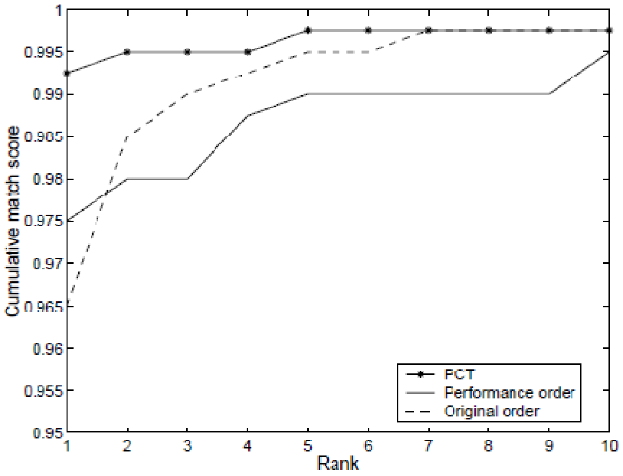


Figure 2.2-14: Performance comparison using 3 bands across all ranks (Pan, Healey, & Tromberg, 2005, p. 148)

Spectral Eigenface

The spectral eigenface method is proposed to solve the conflict between performance and speed (Pan, Healey, & Tromberg, 2005, p. 150). Before processing, it transforms a multiband hyperspectral image to a spectral-face image which samples from all bands recursively while preserving the spatial resolution. Instead of looking at all 31 single-band images one by one (which will increase the complexity of eigenvalue computation due to larger resolution) or resampling the single-band images to lower resolution (which might lose some spatial features), the value of pixel *i* in spectral-face is equal to the value of pixel *i* in band w where *w* is the remainder of *i* divided by 31. The same eigenface technique is applied for face recognition. Figure 2.2-15 illustrates on the right the 10 spectral eigenfaces obtained from all subjects while the image on the bottom left side is the image obtained from the method described above.

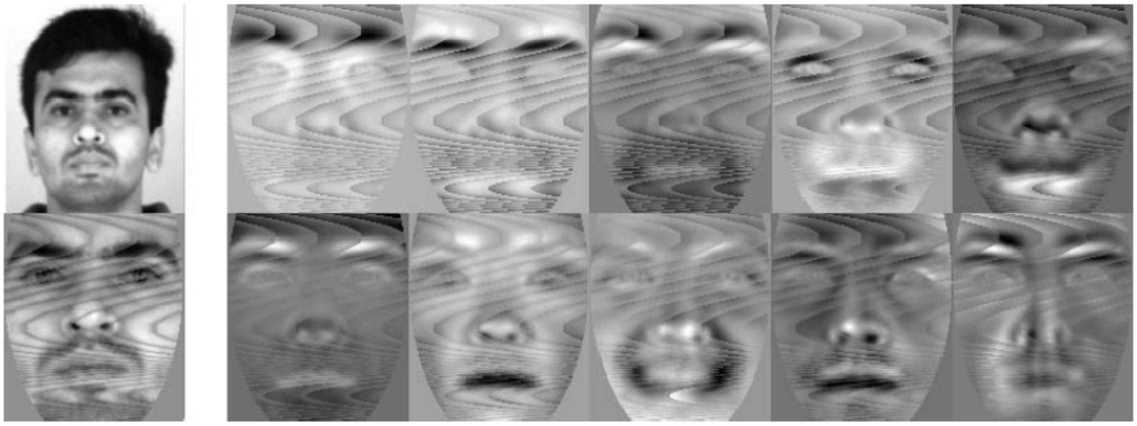


Figure 2.2-15: Spectral-face of a single subject (left) and spectral eigenface obtained from all subjects (Pan, Healey, & Tromberg, 2005, p. 148).

It is shown in Figure 2.2-16 that spectral eigenface performs as well as PCT-based multiband method with much less computation complexity similar to single-band eigenface method.

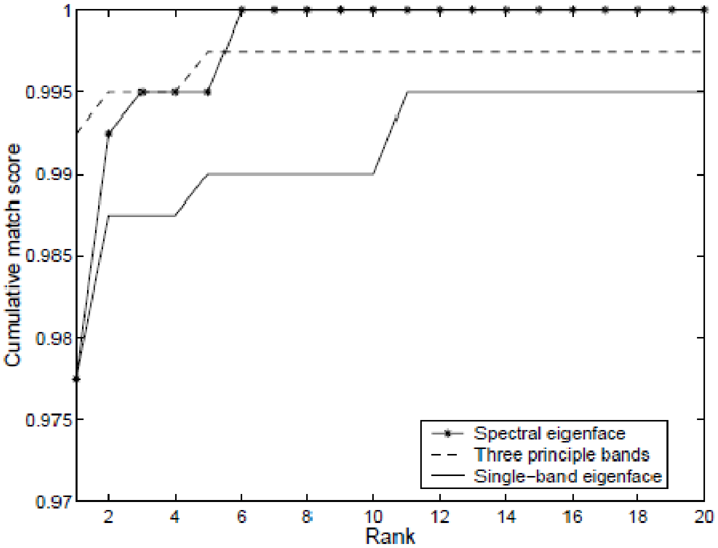


Figure 2.2-16: Performance comparison between Spectral eigenface, Three principle bands, and Single eigenface (Pan, Healey, & Tromberg, 2005, p. 148).

2.3. Scale Invariant Feature Transform (SIFT)

The scale invariant feature transform (SIFT) is an object recognition system that uses local image features. The features are invariant to the scale, translation and orientation of the image and are also partially invariant to illumination changes, affine distortion and change in 3D viewpoint. These features are detected through a staged filtering approach that identifies key points in the scale space. The first stage identifies key locations in scale space by looking for locations that are maxima or minima of a difference-of Gaussian function. Each point is used to generate a feature vector, called SIFT keys, that describes the local image region sampled relative to its scale-space coordinate frame. Partial invariance to affine or 3D projections is achieved by blurring the image gradient locations (Lowe, 1999, p. 1).

Keypoint detection is achieved by identifying locations and scales of the image that can be repeatedly assigned under differing views of the same object. This can be achieved by searching for stable features across all possible scales of the image using a continuous function of scale known as scale space which was first proposed by Witkin (1983). Koenderink (1984) and Lindeberg (1994) have shown that under reasonable assumptions that the only possible scale-space kernel is the Gaussian function. The scale space of an image is then defined as a convolution of a variable-scale Gaussian, $G(x, y, \sigma)$, with image, $I(x, y)$, which can also be defined as a function, $L(x, y, \sigma)$, such that:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y)$$

where

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}.$$

Lowe (1999) proposed using scale-space extrema in the difference-of-Gaussian function convolved image, $D(x, y, \sigma)$, which can also be computed from the difference of two scales separated by a constant factor k (Lowe, 2004):

$$D(x, y, \sigma) = \big(G(x, y, k\sigma) - G(x, y, \sigma)\big) * I(x, y)$$

$$= L(x, y, k\sigma) - L(x, y, \sigma).$$

The initial image is incrementally convolved with Gaussians to produce images separated by a constant factor k in scale space, as shown in the left column in Figure 2.3-1. Adjacent image scales are then subtracted to produce the difference-of-Gaussians images shown on the right column in Figure 2.3-1. Once an octave is completed, the second Gaussian image from the bottom of the stack is down-sampled and the process repeated.

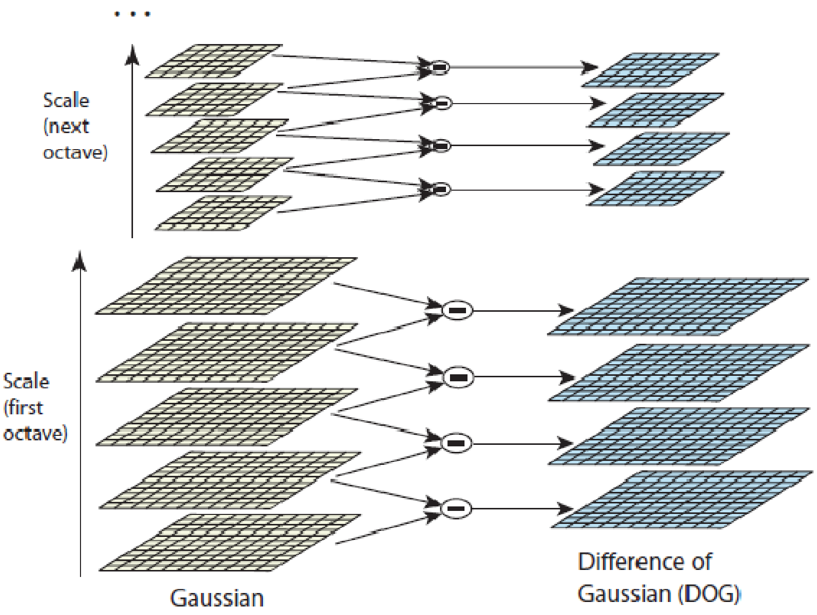


Figure 2.3-1: For each octave of scale space, the initial image is repeatedly convolved with Gaussians to produce the set of scale space images shown on the left. Adjacent Gaussian images are subtracted to produce the difference-of-Gaussian images on the right (Lowe, 2004, p. 6).

Local extrema of $D(x, y, \sigma)$ are detected by comparing each pixel in $D(x, y, \sigma)$ to their eight neighboring pixels in the current scale and the nine neighboring pixels in the scale above and below and they are only selected if they are larger or smaller than all of the compared neighbors. Since most sample points are eliminated after the first few checks, the cost of this process is reasonably low (Lowe, 2004, p. 7).

Once a keypoint candidate has been found by the aforementioned steps, it is then fitted to nearby data for location, scale, and ratio of principal curvatures which allow for the rejection of points that have low contrast or are poorly localized along an edge. This is achieved by fitting a 3D quadratic function to the local sample points to determine the interpolated location of the maximum which is performed by using the Taylor expansion of the scale-space function, $D(x, y, \sigma)$, shifted so that the origin is at the sample point (Lowe, 2004, p. 10):

$$D(\boldsymbol{x}) = D + \frac{\partial D^T}{\partial \boldsymbol{x}}\boldsymbol{x} + \frac{1}{2}\boldsymbol{x}^T\frac{\partial^2 D}{\partial \boldsymbol{x}^2}\boldsymbol{x}$$

Where D and its derivatives are evaluated at the sample point and $\boldsymbol{x} = (x, y, \sigma)^T$ is the offset from this point. The location of the extremum, $\widehat{\boldsymbol{x}}$, is determined by taking the derivative of this function with respect to $\boldsymbol{x}$ and setting it to zero, giving

$$\widehat{\boldsymbol{x}} = -\frac{\partial^2 D^{-1}}{\partial \boldsymbol{x}^2}\frac{\partial D}{\partial \boldsymbol{x}}.$$

Substituting $\widehat{\boldsymbol{x}}$ into $D(\boldsymbol{x})$ gives a function value at the extremum

$$D(\widehat{\boldsymbol{x}}) = D + \frac{1}{2}\frac{\partial D^T}{\partial \boldsymbol{x}}\widehat{\boldsymbol{x}},$$

which is useful for rejecting unstable extrema with low contrast.

Lowe also stated that the difference-of-Gaussian function will have a strong response along the edges. A poorly defined peak in the difference-of-Gaussian function has a large principal curvature across the edge but a small one in the perpendicular direction. These principal curvatures can be computed from a 2x2 Hessian matrix,

$$H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix},$$

computed at the location and scale of the keypoint with derivatives estimated by taking differences of neighboring sample points as suggested by Brown (2002). The eigenvalues of H are proportional to the principal curvatures of D. Therefore, we can determine whether or not a keypoint is along the edges by comparing the ratio of the principal curvatures to some threshold and eliminate as necessary.

A consistent orientation is then assigned to each keypoint based on local image properties so each keypoint descriptor can be represented relative to this orientation and achieve invariance to image rotation. This is performed by using the scale of the keypoint to select the Gaussian smoothed image, L, with the closest scale. This image is then used to compute the gradient magnitude, $m(x, y)$, and orientation, $\theta(x, y)$, by

$$m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2}$$

$$\theta(x,y) = \tan^{-1}\big((L(x,y+1) - L(x,y-1))/(L(x+1,y) - L(x-1,y))\big).$$

An orientation histogram with 36 bins covering 360 degrees is then formed using the orientation of neighboring points with respect to the keypoint. Each sample added to the histogram is weighted by its gradient magnitude and by a Gaussian-weighted circular window with a sigma that is 1.5 times the scale of the keypoint. The highest peak in the histogram is detected and assigned as the dominant directions of local gradient, and any other local peak that is within 80% of the highest peak is used to also create a keypoint with that orientation.

Once the orientation histogram has been formed, a keypoint descriptor is created by first computing the gradient magnitude and orientation at each sample point in a region around the keypoint location which are weighted by a Gaussian window. The samples are then collected into orientation histograms summarizing the contents over n x n subregions as demonstrated in Figure 2.3-2 where the length of the arrows corresponds to the sum of the gradient magnitudes near that direction within the region. Figure 2.3-2 demonstrates a 2x2 descriptor array computed from four 4x4 subregions, but the size of the descriptors and subregions can vary. The descriptor is then modified to reduce the effects of illumination change by normalizing each feature vector to unit length, and to reduce the influence of large gradient magnitudes, caused by illumination changes on 3D orientation, by thresholding the values in the unit feature vector to each be no longer than 0.2 and then renormalizing to unit length.

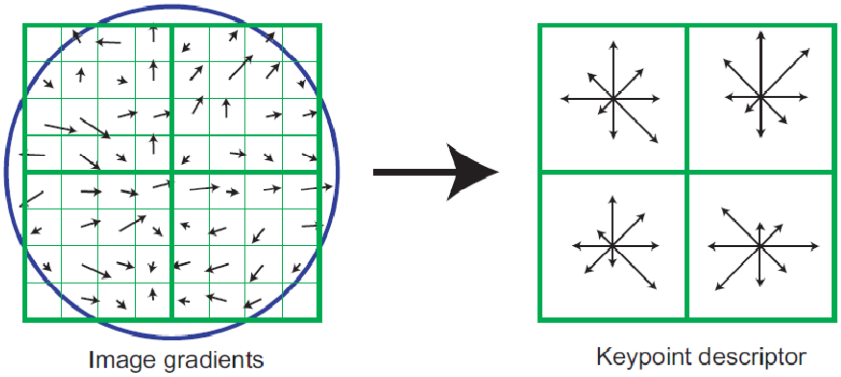


Figure 2.3-2: Illustration of a SIFT descriptor (Lowe, 2004, p. 15).

The SIFT descriptors from each image are then used in a nearest-neighbor approach to indexing to indentify candidate object models. Collections of descriptors that agree on a potential model pose are first identified through a Hough transform hash table, and then through a least-squares fit to a final estimate of model parameters. When at least 3 descriptors agree on the model parameters with low residual, there is strong evidence for the presence of the object. Since there may be dozens of SIFT descriptors in the image of a typical object, it is possible to have substantial levels of occlusion in the image and yet retain high levels of reliability.

An application of SIFT for facial recognition was performed by Luo (2007) using the FERET images with promising result despite performing somewhat poorly on duplicate images (as defined in 2.2.2) compared to some selected algorithms. Our research uses an implementation of a variation of Lowe's SIFT method created by Vedaldi (2006) which is the same version utilized by Ryer et al. (2011) for their research on contextual hyperspectral face recognition.

2.4. Autocorrelation

Similar to Principal Component Analysis, studying the correlation between each observation of a dataset allows us to reduce the dimensionality of said dataset without having a significant loss of information as a side effect. If two consecutive observations of a dataset are highly correlated, then if we analyze the nth and n+1th observations, say O_n and O_{n+1}, very little information will be learned from the second observation since it is closely related to the first observation. On the other hand, if the dataset consists of only independent observations or near independent observations, then O_n has no relationship with O_{n+1}, and more information will be gained from the dataset. If we know each observation is somewhat correlated to its neighboring observations, we can then analyze the autocorrelation between O_n and O_{n+k} for all $k \leq N/4$, where N is the number of observations in the dataset and k is defined as the lag, defined by

$$\rho(k) = Cor[O_n, O_{n+k}].$$

$\rho(k)$ can then be plotted against k to graphically choose the point at which there is zero correlation between two observations as demonstrated in Figure 2.4-1. Welch (1983) suggests that roughly 95% of the correlation should be removed by choosing k to be the point at which $\rho(k)$ drops below D or rises above -D, where D is defined as:

$$D = 2\left(\sum\nolimits_{k=-L}^{L} \frac{\hat{\rho}(k)}{N}\right)^{1/2}.$$

Figure 2.4-2 shows the autocorrelation function dips below D at a lag of thirty two observations which suggests that approximately 95% of the original information can be retained by choosing every thirty-second, or less, observations (Welch, 1983, p. 306).

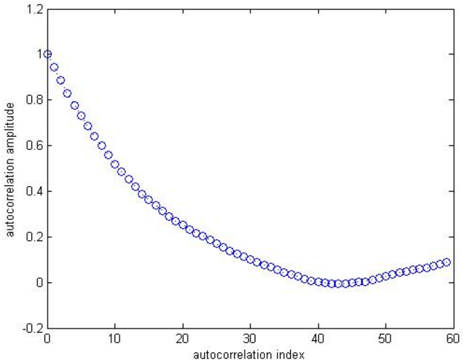


Figure 2.4-1: Autocorrelation between observations.

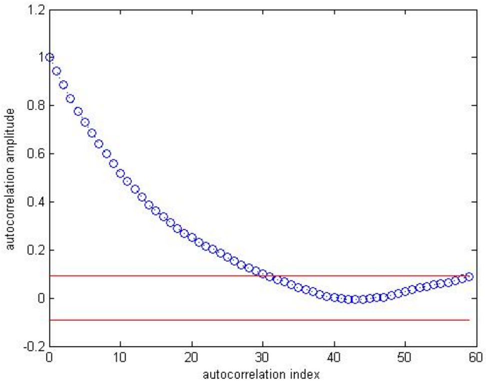


Figure 2.4-2: Welch's D (red line) removes 95% of correlation.

Williams (2007) used correlation based data reduction while clustering pixels in hyperspectral images to detect anomalies in the scene. Williams showed that the performance degradation was minimal despite only using every fifth band in the datacube.

2.5. One-Quarter Fractional Factorial Design of Experiment

Montgomery (2009) describes a 2^k factorial design as a design of experiment that is consisted of k number of factors or parameters that can all be set to at two levels of low and high. The model of a 2^k includes k main effects, $\binom{k}{2}$ two-factor interactions, $\binom{k}{3}$ three-factor interactions, and up to a single k-factor interaction. However, as the number of factors in a 2^k factorial design increases, the number of design points required to perform a complete replicate of the design grows exponentially and may outgrow the resources available to run the experiment. For example, a complete replicate of a 2^6 design requires 64 runs, and only 6 of the 63 degrees of freedom correspond to the main effects, i.e. the six factors, and 42 degrees of freedom correspond to three-factor and higher interactions. Therefore, if one can reasonably assume that certain high-order interactions are negligible, information on the main effects and low-order interactions may be obtained by running only a fraction of the complete factorial experiment by conduction a fractional factorial design. Such designs are among the most widely used types of designs for product and process design and for process improvement (Montgomery, 2009, p. 289).

One of the major uses for fractional factorials is in screening experiments where many factors are considered and the objective is to identify those that have large effects.

These are usually performed in the early stages of a project when many of the factors initially considered likely have little or no effect on the response. The factors identified as important are then investigated more thoroughly in follow up experiments (Montgomery, 2009, p. 290). For the purpose of this thesis, we will only consider the one-quarter fraction of the 2^k design since we will be dealing with a moderately large number of factors with a considerably long runtime associated with each design point.

A one-quarter fraction of the 2^k design contains 2^{k-2} runs and is constructed by first creating a basic design consisting of the runs associated with a full factorial in k-2 factors. The two additional factors are added by assigning their plus and minus levels to the plus and minus levels of the interactions chosen from the first k-2 factors. Table 2.5-1 demonstrates the construction of a one-quarter fraction of a 2^6 design. As we can see, the interactions ABCE and BCDF yield columns of plus levels or the identity column I, so ABCE and BCDF are therefore defined as the 'generators' of this particular design. We can also verify that every main effect is aliased by three- and five-factor interactions, whereas two-factor interactions are aliased with each other and with higher order interactions. Therefore, when we estimate A, we are actually estimating A + BCE + DEF + ABCDF. Table 2.5-2 provides a complete alias structure of this design. As previously stated, if three-factor and higher interactions are negligible, this design gives clear estimates of the main effects. It should be noted that this design is also denoted as a Resolution IV design since no main effect is aliased with any other main effect or with any two-factor interaction, but two-factor interactions are aliased with each other.

Table 2.5-1: 2^{6-2} RES IV Fractional Factorial design.

	Basic Design					
Run	A	B	C	D	E = ABC	F = BCD
1	-	-	-	-	-	-
2	+	-	-	-	+	-
3	-	+	-	-	+	+
4	+	+	-	-	-	+
5	-	-	+	-	+	+
6	+	-	+	-	-	+
7	-	+	+	-	-	-
8	+	+	+	-	+	-
9	-	-	-	+	-	+
10	+	-	-	+	+	+
11	-	+	-	+	+	-
12	+	+	-	+	-	-
13	-	-	+	+	+	-
14	+	-	+	+	-	-
15	-	+	+	+	-	+
16	+	+	+	+	+	+

Table 2.5-2: Aliasing structure

A = BCE = DEF = ABCDF	AB = CE = ACDF = BDEF
B = ACE = CDF = ABDEF	AC = BE = ABDF = CDEF
C = ABE = BDF = ACDEF	AD = EF = BCDE = ABCF
D = BCF = AEF = ABCDE	AE = BC = DF = ABCDEF
E = ABC = ADF = BCDEF	AF = DE = BCEF = ABCD
F = BCD = ADE = ABCEF	BD = CF = ACDE = ABEF
	BF = CD = ACEF = ABDE
ABD = CDE = ACF = BEF	
ACD = BDE = ABF = CEF	

Once the experiment is completed and the respective responses obtained, all of the model adequacy checks, statistical analysis, and estimation of model parameters associated with a full 2^k factorial design can be performed. Estimates of the effects can be used to get an idea of which factors influence the model which can then be followed by an analysis of variance (ANOVA) to determine whether or not any of these factors or interaction of factors is significance.

2.6. Ensemble Classifications

Suppose there are multiple classifiers for a single classification problem, one method of utilizing such situation is to fuse the results obtained from the various classifiers into one single assignment. Polikar (2006) demonstrates a handful of methods that are widely used in fusing a set of classifiers into one ensemble decision which include the Mean Rule, the Minimum/Maximum/Median Rule, the Product Rule, and the Borda Count. The first five methods of fusion work just as their name suggests where the overall support, ω_j, for each class j is obtained by the listed fusion rule, $\Im(.)$, across the T number of classifiers

$$\omega_j(\boldsymbol{x}) = \Im\left[d_{1,j}(\boldsymbol{x}), \dots, d_{T,j}(\boldsymbol{x})\right]$$

where $d_{i,j}(\boldsymbol{x})$ is the individual support for $\boldsymbol{x}$ for class j obtained from the i-th classifier. Assignment is then made to the class with the highest $\omega_j(\boldsymbol{x})$.

One method that is typically used when the classifiers can rank order the classes is the Borda count method which was originally devised in 1770 by Jean Charles de Borda. This can be performed if the output provided by the classifiers is continuous since the classes can then be rank ordered with respect to the output score given by the classifier. Borda count only needs the rankings and not values of these continuous outputs; hence it qualifies as a combination rule that apply to labels. The standard Borda count requires each classifier to rank-order the classes. If there are *N* classes, the first-place class is given *N*-1 votes followed by the second ranked class with *N*-2 votes, and this continues all the way down to the last ranked class with 0 votes. The votes are then summed across all classifiers and the class with most votes is than chosen as the fused decision.

The Borda count method is used widely in a variety of applications including but not limited to: selecting the most valuable player in U.S baseball league, ranking universities in college sports, electing officers at certain university senate elections, and choosing the winning song in the annual European wide song contest in Eurovision, all use some variation of Borda count (Polikar, 2006).

2.7. Diversity Measures

Polikar (2006) stated that if there is access to a classifier with perfect generalization performance, there would be no need to resort to ensemble techniques, but the presence of noise, outliers and overlapping data distributions, however, make such a classifier an impossible proposition. Polikar then continued by stating that

> "The strategy in ensemble systems is therefore to create many classifiers, and combine their outputs such that the combination improves upon the performance of a single classifier. This requires, however, that individual classifiers make errors on different instances." (Polikar, 2006, p. 24)

The simplest form of diversity measures are pair-wise measures which are defined between two classifiers, and for *T* number of classifiers, we calculate $\frac{T(T-1)}{2}$ unique pair-wise measures. Given two hypotheses h_i and h_j, we define the following notation

	h_j *is correct*	h_j *is incorrect*
h_i *is correct*	a	b
h_i *is incorrect*	c	d

where a is the proportion of instances that are correctly classified by both h_i and h_j, b is the proportion of times that h_i correctly classified but h_j incorrectly classified, c is the proportion of times that h_j correctly classified but h_i incorrectly classified, and d is the proportion of times that both h_i and h_j incorrectly classified.

We can then use the information obtained to calculate the pair-wise diversity measures such as the Diversity Correlation, Q-Statistic, Disagreement and Double Faults measures. The Diversity Correlation is defined as

$$\rho_{i,j} = \frac{ad - bc}{\sqrt{(a+b)(c+d)(a+c)(b+d)}} \quad -1 \leq \rho \leq 1,$$

and the Q-Statistic measure is defined as

$$Q_{i,j} = \frac{ad - bc}{ad + bc}$$

where maximum diversity is obtained either $\rho = 0$ or $Q = 0$. Disagreement is the proportion of times that either classifier misclassified when the other correctly classified whereas Double Faults are the proportion of times that both classifier misclassified and they are defined, respectively, as

$$D_{i,j} = b + c$$

and

$$DF_{i,j} = d.$$

Non pair-wise diversity measures can also be calculated given the information above which include the Entropy and Kohavi-Wolpert Variance defined, respectively, as

$$E = \frac{1}{N}\sum_{t=1}^{N} \frac{1}{T - \lceil T/2 \rceil} \min\{\zeta_{\mathrm{t}}, (\mathrm{T} - \zeta_{\mathrm{t}})\}$$

and

$$KW = \frac{1}{NT^2}\sum_{t=1}^{N} \zeta_t(\mathrm{T} - \zeta_t)$$

where *N* is the number of observations in the dataset, ζ_t is the number of classifiers that incorrectly classified observation $\boldsymbol{x}_t$, and both measure has a range between 0 and 1 where 0 indicates no diversity and 1 indicates maximum diversity.

2.8. Carnegie Mellon University Hyperspectral Face Database

Denes, Metes, and Liu (2002) from the Robotics Institute at Carnegie Mellon University began their hyperspectral face images collection in October 2001 covering the spectral range between 450 nm to 1100 nm. As of October 2002, the date at which the data being used for this research is published, the database contains 54 diverse faces at multiple sessions over a period of about two months. The data was collected using a limited performance (sic) prototype CMU-developed spectro-polarimetric camera.

The Spectro-polarimetric imaging camera covers a spectral range of 450 to 1100 nm with a spectral band pass of 10 nm. The polarimetric capabilities of the camera were not used for the data collection. The camera control software was written in Visual Basic which controls the spectral filtering hardware and is operated from a desktop computer. An external frame grabber software of the medium resolution (640 x 480) analog camera handles the frame acquisition and presentation functions to the desktop computer. Figure 2.6-1 shows the spectro-polarimetric imaging camera specifications.

<table>
<tr><td colspan="2"> </td></tr>
<tr><td colspan="2"><u>Specifications</u></td></tr>
<tr><td>AO material</td><td>TeO_2</td></tr>
<tr><td>Spectral range</td><td>450-1100 nm</td></tr>
<tr><td>Resolution</td><td>10 nm @600nm</td></tr>
<tr><td>AO efficiency</td><td>25-70 MHz</td></tr>
<tr><td>Retarder range</td><td>400-1800 nm</td></tr>
<tr><td>IFOV</td><td>~7deg</td></tr>
<tr><td>RF power</td><td><1 W</td></tr>
<tr><td>AOTF aperture</td><td>15 x 15 mm</td></tr>
<tr><td>AO Interaction</td><td>15 mm</td></tr>
<tr><td>Crystal length</td><td>26.5 mm</td></tr>
<tr><td>Min. illumination</td><td>CCD Camera dependent</td></tr>
</table>

Figure 2.8-1 (Denes, Metes, & Liu, 2002, p. 5)

Data collection of the 54 subjects took place between 10/18/01 and 12/4/01 after which data collection was suspended due to feedback from test subjects indicating that the photo lamps were causing some residual eye irritation during and shortly after exposure (Denes, Metes, & Liu, 2002). The dataset is consisted of 5 image sessions where each session consists four multispectral images where illumination is varied as followed: 1) lights at 45 degree right on, 2) center light only, 3) lights at 45 degree left only, and 4) all lights on. Each multispectral image, or datacube, covers the spectral range between 450 to 1100 nm in steps of 10 nm resulting in 65 spectral bands. The session breakdown of the 54 subjects is as followed: 54 subjects participated in the first session. However, only 36 of the 54 subjects returned for a second session, 28 of the 36 returned for the third session, 22 of the 28 returned for the fourth session, and 16 of the 22 returned for the fifth session. For our research, the first session will be assigned as the training set and the second session as the testing set. The additional sessions will be used to study the effect of temporal changes. Figure 2.6-2 illustrates the effect of the illumination variation over five different spectral bands (550 nm, 650nm, 750 nm, 850 nm, and 1000 nm).

One problem with the data obtained is the fact that we observe darkened, noisy exposures at the low and high end of the spectrally filtered images. The Exposure of the CCD camera is set at its fixed maximum value of 1/60 sec. Lighting and CCD has its maximum response around 650nm. At shorter and longer wavelengths, there are not enough photons to produce noise-free images although it is not specified why this is true (Denes, Metes, & Liu, 2002).

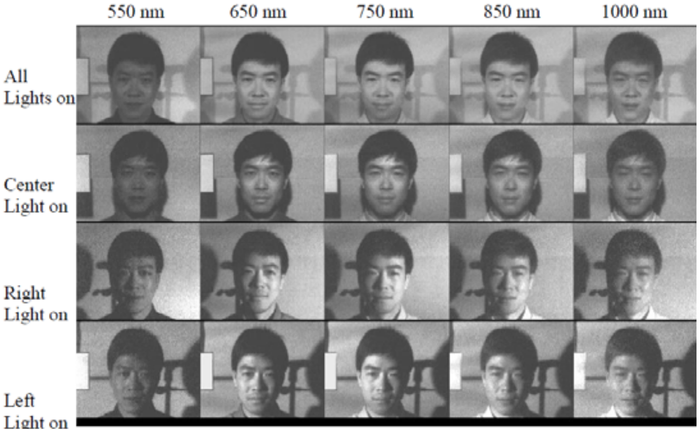


Figure 2.8-2 (Denes, Metes, & Liu, 2002, p. 10)

3. Methodology

In this chapter, we outline the methodology and processes undertaken to perform our research using the concepts reviewed in chapter 2. We first study the correlation between the spectral bands of all images in the target set. This is followed by applying SIFT on the least correlated bands and matching the descriptors of a set of probe images with that of the target images. A designed experiment is then conducted to determine the significance of the adjustable parameters in the algorithms and to obtain an optimal setting that maximizes the accuracy of the matching algorithm. This is followed by SIFT matching of the test images and the train images using the optimum settings obtained from the designed experiment as well as SIFT matching on the eigenfaces obtained through PCT of the two image sets. The ensemble performances of independent spectral bands and eigenfaces are then considered to achieve a possible dominating classification. All matching performances are then evaluated using the method proposed by FERET.

3.1. Study on Correlation between Bands

Similar to the work peformed by Williams (2007), skin components within the datacube is studied to see if there is any correlation within the hyperspectral bands. To perform this study, the skin component within each band has to first be extracted from the datacube and this is done by using the poorman_skin_detection.m function by Ryer et al. (2011) that utilizes the Normalized Differential Skin Index (NDSI) of the hyperspectral images as proposed by Nunez (2009). Once this is completed, each pixel will then be treated as one single run of 55 observations, as defined in 2.4, comprised of the intensities of bands 6-60. The reason for removing the first and last five bands of the datacube is because the images from these ranges are noisy as described in 2.4. The result is then a two dimensional dataset with pixel count of rows and 55 columns. Since only the skin component of the datacube is extracted by the poorman_skin_detection.m function, any non-skin pixels will consist of zero-valued observations, so the number of rows can then be reduced by removing any rows whose average rounds down to zero. The final dataset will then have skin pixel count of rows and 55 columns

The number of rows in the final dataset is still somewhat large and falls around the magnitude of 10^4. However, initial exploratory runs indicated inter-band correlation does not change after resizing the image resolution down to 120 x 160 from the original 480 x 640 (number of rows is subsequently reduced by one magnitude) which results in faster computation. Therefore, as a preprocessing step, all training images are resized to 120 x 160 x 65 before computing the average $\rho(k)$ of the skin component for each subject and the average $\rho(k)$ for all subjects in the training dataset. However, all of the following analyses are performed on the original dataset with spatial resolution of 480 x 640, and the reason for doing so is to allow SIFT to pick up as many descriptors as possible. We use the down-sampled image for correlation study just for the convenience since we find from experiments on a few subjects that the autocorrelation of the skin does not vary between the two image sizes.

3.2. SIFT on Uncorrelated Bands

Using the obtained information from 3.1, the next step is to perform target matching using SIFT on the uncorrelated bands. The functions used for our work are sift.m and siftmatch.m for analytical purposes and plotmatches.m function for visualization purposes. The MATLAB® code for all three functions listed are included in Appendix C for the reader's convenience. Auxiliary functions called within the three functions are not included but can be obtained directly from the author's website. The sift.m function extracts the SIFT frames FRAMES and their descriptors DESCR, as defined in 2.3, from an image. FRAMES is a 4 x *k* matrix storing one SIFT frame per column where:

FRAMES(1:2, k) is the center (X,Y) of the frame *k*,

FRAMES(3, k) is the scale SIGMA of the frame *k*,

FRAMES(4, k) is the orientation THETA of the frame *k*.

DESCR is a D x *k* matrix that stores one descriptor per column (usually D=128). The siftmatch.m function will then match the descriptors from the two images that are to be matched at a specified threshold. The function uses the same algorithm suggested by Lowe (2004) to reject poor matches. A descriptor DESCR1 is matched to a descriptor DESCR2 only if the distance d(D1,D2) multiplied by THRESH is not greater than the distance of D1 to all other descriptors. The function then returns MATCHES that is a 2 x *m* matrix where *m* is the number of possible frame matches and $\text{MATCHES}_{1,\,m}$ is a row of SIFT frames from the first image that is matched to the SIFT frame of the second image $\text{MATCHES}_{2,\,m}$.

The idea is to then use the number of possible matches, *m*, as a matching score for a single pair of bands. Since there are multiple band-comparisons for each pair of subjects, we can use the various ensemble methods described in 2.6 with the available band scores and assign the highest ensemble matching to be the true match. Our function, hsi_sift_match.m, takes in five variables as input: target_image, gallery_image, eraser_size, match_threshold, and lag. Eraser_size is called by the poorman_skin_detection.m function that extracts only the pixel values associated with the skin feature within an image cube and assigns other pixel values to zero. It has an interval of $[0, \infty)$ where an increase in the parameter results in a decrease in the number of skin pixels. The match_threshold parameter is called by the siftmatch.m function that performs a matching between the descriptors associated with two images as extracted by the sift function. It also has an interval of $[0, \infty)$ where increase in the parameter value reduces the number of possible descriptor matches between the two images. The lag parameter is a step size that hsi_sift_match.m will use in selecting the bands to be passed through sift.m for descriptors extraction. It then gives out the number of possible SIFT descriptor matches, *m*, as an output. Figure 3.2-1 illustrates a matching of descriptors within the train and test images of a subject at band 22 using the default settings for sift.m and siftmatch.m and settings of 65 for eraser_size. The number of lines is the number of matches, *m*, as described above.

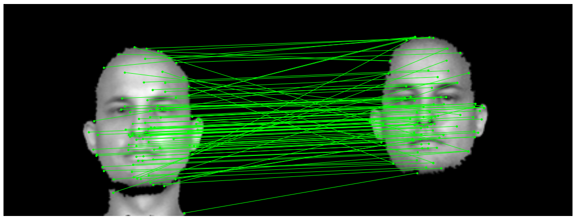


Figure 3.2-1: Example of SIFT matching for one subject.

3.3. Fractional Factorial Design on Settings

A design of experiment is then required in order to find the optimum settings for hsi_sift_match.m or to see whether any of the parameters used in the function are insignificant and can therefore be set to its least computationally intensive setting. Since there are three adjustable parameters in the hsi_sift_match.m function that affects the accuracy and computational speed of the matching process, i.e. eraser_size, match_threshold, and lag; these parameters will then be assigned as possible factors in the experimental design.

The computing time of a single matching iteration of hsi_sift_match.m has been shown to be extremely lengthy especially if real time application is desired. However, it was discovered through experimentation that three parameter adjustments could be made in the sift.m function that could decrease the time by about 10-15 seconds, which equates to a 32-48% improvement to the runtime. Therefore, the effects that these parameters have on the performance of the algorithm is also part of our study. The parameters mentioned above are NumLevels, EdgeThreshold, and Magnif and are internal parameters within the sift.m function. NumLevels is the number of scale levels within each octave. Features that have a flatness score above EdgeThreshold are ignored and bigger EdgeThreshold values imply more features accepted. Magnif is the frame magnification factor, and each spatial bin of the SIFT histogram has an extension equal to (magnif x σ), where σ is the scale of the frame.

The 2_{IV}^{6-2} fractional factorial design as described in 2.5 is then used where eraser_size, match_threshold, lag, and numLevels are assigned as the factors for the initial basic design, and are denoted as A, B, C, and D, respectively. The levels for EdgeThreshold and Magnif are assigned using ABCE and BCDF as the generators, respectively. These variables will be referred to as Eraser, Match, Lag, Scales, Edge, and Frame, respectively, for the purpose of the Design of Experiment which have the following low and high settings as listed in Table 3.3-1. The resulting design is identical to the one shown in Table 2.5-1 with 16 total runs where a single run is defined as matching the 36 subject images taken during the second session against the images of the same 36 subjects taken during the initial session.

Table 3.3-1: Low and High settings of each factors.

Parameter	Low Setting	High Setting
Eraser	5	65
Match	3	7
Lag	16	18
Scales	2	3
Edge	5	10
Frame	2	3

The purpose for running the designed experiment is to determine if any of the mentioned parameters are insignificant with respect to the matching performance of each band. Any factors deemed insignificant are then set to the "fast" setting to reduce the computation time without a lost in performance (a fast setting for Match and Lag would be the larger value since they take away more information from the datacube and allow for faster computation). It should be mentioned that, intuitively, the "slow" settings for all parameters results in the best performance since they allow for most information to be included.

3.4. SIFT on PCTs with Optimized Settings

For the remaining of this paper, the term PCTs will refer to the principal bands of the subject's hyperspectral image obtained using PCT as suggested by Pan (2005). Intuitively, applying SIFT to the first few PCTs of a datacube should give a similar result to that obtained from uncorrelated bands since the principal components are orthogonal and each component provides information that the others do not (i.e. they are uncorrelated). The idea is to then find out if similar or better results can be obtained by SIFT-PCT using a fewer number of PCTs than the number of uncorrelated bands it takes in the method described in 3.3 since this would require less iterations of SIFT and is computationally quicker.

The PCTs of each subject are extracted from their respective skin-reduced datacube that is obtained similarly to the method in 3.1 using the poorman_skin_detection.m function. The resulting datacube is then converted into a 2 dimensional array of size (480 x 640) x 65 where each row accounts for the intensities of a single pixel across all 65 bands. The principal components are then obtained from the covariance matrix by first finding its eigenvectors and then pre-multiplying the reshape data with each eigenvector. Each principal component accounts for its associated eigenvalue over the sum of eigenvalues of the total variance across all bands and can be treated as an orthogonal image of the particular subject. If a large percentage of the total variance can be captured with a smaller number of principal components than the number of bands it takes in 3.3, then we realize a computational cost reduction associated with the reduction in required comparisons.

The eigenface.m function performs the processes above by converting an input face datacube into a principal component datacube output. The number of components selected by eigenface.m is based on the Max Euclidean Distance from Log-Scale Secant Line method as proposed by Johnson (2008) for locating the “knee” in the eigenvalues curve. The time it takes to obtain the principal components of each image should be considered since the process requires a considerable number of computations. Therefore calculating the PCTs of each subject as a preprocess step and passing in the PCTs instead of the datacube through the hsi_sift_match.m function would be preferred option. If the result obtained from the PCTs is better than that obtained from the original datacube, then this would be a desirable tradeoff.

3.5. Ensemble of Uncorrelated Bands/PCTs Matching

Since we perform multiple band-comparisons in 3.2 and 3.4, we essentially have multiple classifications and multiple decisions, possibly conflicting, that can be combined to give us an ensemble decision. As mentioned in 2.6, five available methods that could be used to perform such task are the Sum Rule, Mean Rule, Minimum Rule, Maximum Rule, Median Rule and Borda Count where each function is applied to the support/score provided by each classifier which is then used as a decision for assignment.

The hsi_sift_match.m function outputs the number of possible descriptor matches between the probe and target datacubes as an array of 1 x (number of bands). We first unitize the output for a particular *Band* between probe $\boldsymbol{x}$ and target *j* by dividing each score, $s_{Band\ i,j}(\boldsymbol{x})$, with the sum of scores that probe $\boldsymbol{x}$ obtained over a set of 36 targets as follows

$$d_{Band\ i,j}(\boldsymbol{x}) = \frac{s_{Band\ i,j}(\boldsymbol{x})}{\|s_{Band\ i}(\boldsymbol{x})\|},$$

where

$$\|s_{Band\ i}(\boldsymbol{x})\| = \sum_{j=1}^{36} s_{Band\ i,j}(\boldsymbol{x})\,;$$

which is performed so that the scores of the various classifiers are transformed into a uniform scale. The overall decision profile, *DP(**x**)*, of probe $\boldsymbol{x}$ can then be represented as the following matrix

$$DP(\boldsymbol{x}) = \begin{bmatrix} d_{Band\ 1,1}(\boldsymbol{x}) & \cdots & d_{Band\ 1,j}(\boldsymbol{x}) & \cdots & d_{Band\ 1,36}(\boldsymbol{x}) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ d_{Band\ i,1}(\boldsymbol{x}) & \cdots & d_{Band\ i,j}(\boldsymbol{x}) & \cdots & d_{Band\ i,36}(\boldsymbol{x}) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ d_{Band\ T,1}(\boldsymbol{x}) & \cdots & d_{Band\ T,j}(\boldsymbol{x}) & \cdots & d_{Band\ T,36}(\boldsymbol{x}) \end{bmatrix}$$

where T is the number of uncorrelated bands and j is the index of the subject in the gallery. Using Sum, Mean, Max, Min, and Med rule, the ensemble support can then be obtained by applying the respective function, $\Im(.)$, to the support for each class j by

$$\omega_j(\boldsymbol{x}) = \Im[d_{Band\ 1,j}(\boldsymbol{x}), \dots, d_{Band\ T,j}(\boldsymbol{x})]$$

and assigning target $\boldsymbol{x}$ to be subject j that has the highest $\omega_j(\boldsymbol{x})$.

Using the Borda count method, $d_{Band\ i,j}(\boldsymbol{x})$ can then be used to rank the likelihood that a subject j is the same person as the target $\boldsymbol{x}$. For this research, the number of classes is equal to the number of subjects in the gallery and the number of classifiers is equal to the number of band/PC-comparisons. For each target, a “Borda count” between 0 and 35 will then be assigned to the lowest up the highest “SIFT score” gallery matches, respectively. This is repeated for each band/PC and the “Borda counts” from each band/PC is tallied for each subject in the gallery and the gallery-subject with the highest Borda count will be assigned as a match for that particular target.

We will ultimately look at the diversity between the bands and PCTs to gain some insights on the relationship between the bands and their performances and perform an ensemble of the best performing individual pieces to obtain a dominating classifier, if possible.

3.6. Evaluation by FERET

Pan et al. (2003) mimic the FERET evaluation despite not being able to compare their results directly since the data being used in each study were different; grayscale images in FERET and hyperspectral images in Pan et al.. However, Pan et al. was able to replicate all types of variations included in the FERET study such as facial expression and pose. We did not have the capability to collect our own set of data and were not able to obtain the one used in Pan's study but were fortunate to obtain the one collected by Carnegie Mellon. Since the dataset does not include pose or expression variation, we cannot truly compare our results with those given by Pan. We do, however, use the same methodology described by the FERET study and used by Pan to evaluate our result, the closed universe evaluation method. However, since the images were taken over a span of two months, we are able to study the robustness of our algorithm with respect to temporal changes as the temporal, spatial, and spectral properties changes slightly.

4. Results and Analysis

4.1. Correlation between Bands

As previously mentioned, the first and last five bands are thrown away prior to the correlation analysis due to the amount of noise within those images. Figure 4.1-1 illustrates the skin correlation between the bands for all 36 subjects in the training set whereas Figure 4.1-2 shows the mean skin autocorrelation of the training set. Based on the latter figure, we can see that the correlation between the bands is close to zero when they are 16 to 18 bands apart. Therefore, given 55 “useful” bands, a lag between 16 and 18 allows us to reduce the hyperspectral dimension from 55 down to only four bands (i.e. band 6, 22, 38, 54 with lag = 16) without much information lost. We can then apply Vedaldi's sift.m function on each of the four images to extract the SIFT descriptors within the face of the subjects instead of using all 55 bands. This expedites the process significantly.

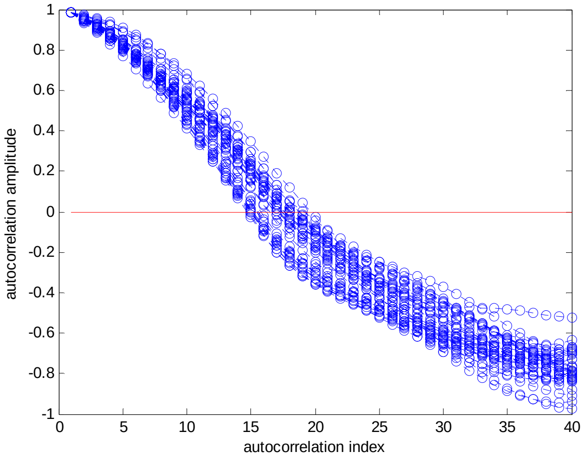


Figure 4.1-1: Autocorrelation of skin pixels across 55 bands of all 36 subjects.

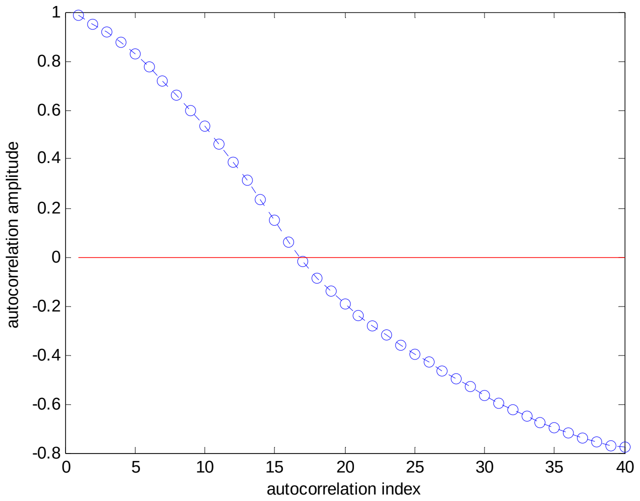


Figure 4.1-2: Mean autocorrelation of skin pixels across 55 bands.

It should be noted that given the three possible lag values above, there are 12 unique combinations of bands that span the 55 bands. However, for the design of experiment conducted in the following section, we only consider the sets that begin with the sixth out of all 65 bands which give us the following three unique band combinations: {6, 22, 38, 54}, {6, 23, 40, 57}, and {6, 24, 42, 60}.

4.2. Design of Experiments on Settings

After performing the experimental design laid out in 3.3 using the matching accuracy of each band as the response variables, it is found that all of the parameters except for Scales are shown to be significant in at least one of the four bands as illustrated by the Half-Normal Plots in Figure 4.2-1.

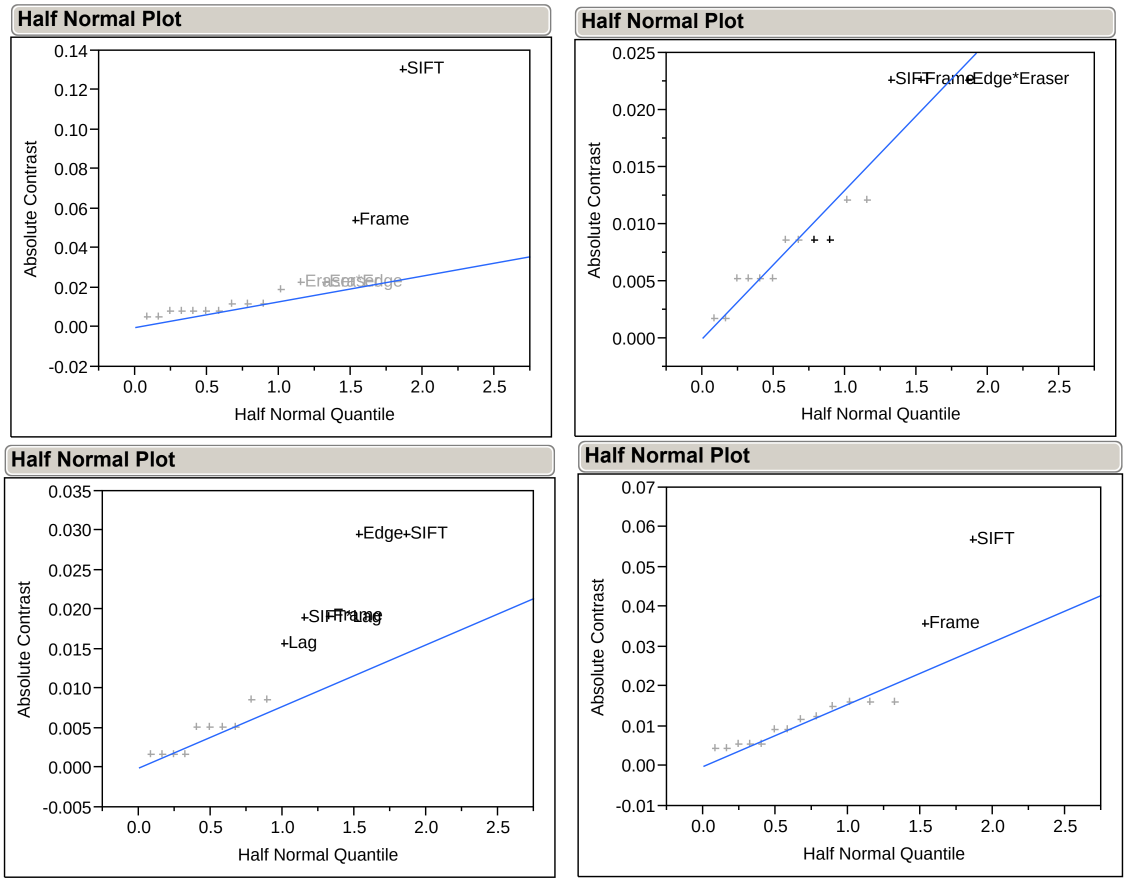


Figure 4.2-1: Band 1 (top left), Band 2 (top right), Band 3 (bottom left), Band 4 (bottom right).

This is supported by the Analysis of Variance (ANOVA) of each band's Design of Experiment where the factors/parameters are shown to have significance in the overall matching accuracy for the respective band. The ANOVA tables and assumption diagnostics are included in Appendix A. In cases where the parameter is shown to be insignificant but the interaction of the parameter is shown to be significant, we include the insignificant parameter in our model to maintain hierarchy. Based on these result, we conclude that the Scales setting in the sift.m function can be set to its fast setting. However, instead of adjusting the remaining five parameters to their slow settings, we instead find the optimum setting that maximizes the accuracy of each band while fixing Scales to its fast value. This is performed using Design Expert®'s Optimization feature by treating the accuracy of each band as objective functions with respect to the significant parameters which are constrained to their respective lower and upper settings. This results in an optimum setting of low for Eraser = 5, Match = 3, Lag = 16, and Scales = 2, and high for Edge = 10 and Frame = 3. Under these settings and taking into account the fact that the first and last five bands are removed prior to performing SIFT matching, Band 1, 2, 3 and 4 spans the spectral wavelength of 500-510nm, 660-670nm, 820-830nm, and 980-990nm, respectively. The matching algorithm was then run again using the optimum setting above which results in matching performance of 58%, 92%, 83%, and 81% for Band 1 through 4 at Rank 1, respectively. The performance for each band across the rank is illustrated in Figure 4.2-2.

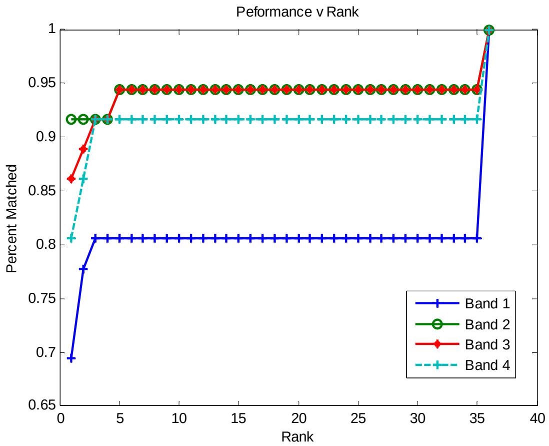


Figure 4.2-2: Performance by bands.

The performance of Band 1 is much lower than that of the other three bands across all rank. The reason for the inability of each band to obtain perfect matching at lower ranks is due to the heavy distortion of the target image of two subjects after performing skin detection which results in zero SIFT matching as shown in Figure 4.2-3. In our implementation of performance ranking, any matches that have a score of zero are penalized and moved to the end of the line up and are not recognized until all other matches are considered.

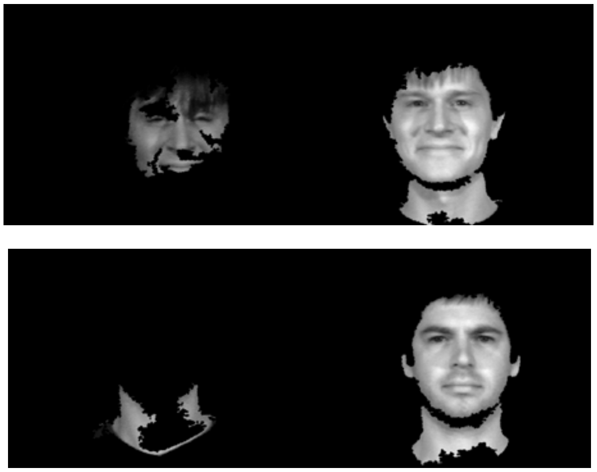


Figure 4.2-3: Band 2 test images (left) of subject A07 (top) and A12 (bottom) are heavily distorted after skin detection.

4.3. SIFT on PCTs

Using the optimal settings obtained from 4.2, we then perform a SIFT matching between the PCTs of the target and probe. This is performed similar to the method used for the uncorrelated bands where each PCT is matched only to its corresponding PCT (i.e. the first PCT of the target is matched with the first PCT of the probe and so on). Based on the Max Euclidean Distance from Log-Scale Secant Line method as proposed by Johnson (2008) in 3.4, it is suggested that six principal components should be retained. However, as previously stated in 3.4, we would like to see if better performance can be obtained by using at most as many bands as the number used in the uncorrelated bands method. We therefore retained only four principal components or PCTs from each datacube and performed SIFT matching using those PCTs.

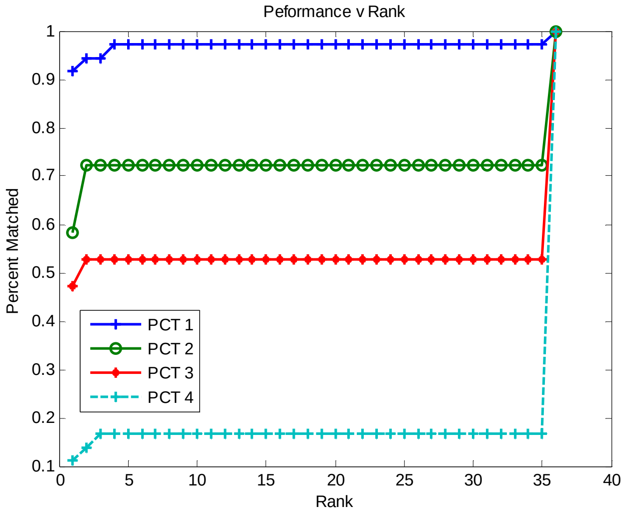


Figure 4.3-1: Performance by PCT.

Figure 4.3-1 shows the performance of each PCTs across all ranks, and similar to the results obtained in 4.2, perfect matching is not obtained until rank 36 due to the distortion to the image of subject A08 after performing skin detection. However, in comparison to the uncorrelated bands method, the performance using PCTs is higher within the first 5 rank where an accuracy of 97.22% is obtained at rank 4 whereas the best single band performance at rank 4 is 94.44% obtained using Band 2. It is shown then, that the PCT method can provide a better performance since most of the variance across all 55 bands are captured by the first four principal components.

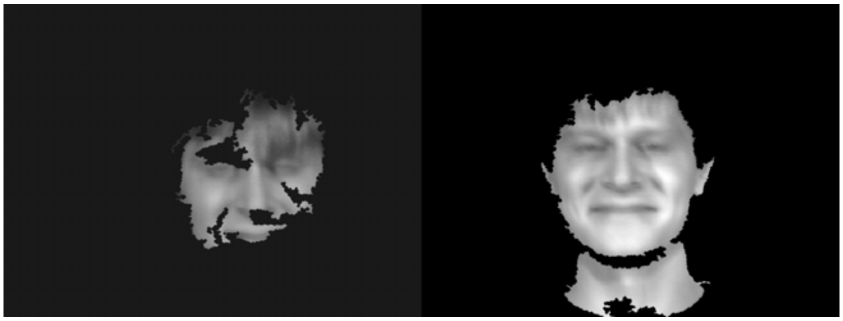


Figure 4.3-2: 1st PCT test image (left) of subject A07 is heavily distorted after skin detection.

4.4. Fusion of Uncorrelated Bands Matching

It is shown in 4.2 and 4.3 that some of the bands and PCTs (specifically Band 2, 3, 4 and 1st PCT) provide performances in the range of 92%-98% within the first 5 rank. We therefore will want to, at the very least, investigate the possibility of obtaining a better performance by combining these individual classifiers using the Sum, Max, Min, Mean, Median rules and Borda count as described in 3.5.

We first perform an ensemble of all eight classifiers using the various methods and compare that to an ensemble of the top four performing classifiers listed above. The Min rule fusion, however, will not be performed on the ensemble of eight classifiers since it will mostly pick up the zero scores given by the 4th PCT and result in a really low performance. Figure 4.4-1 illustrates the performance of each of the ensemble methods using all eight classifiers whereas Figure 4.4-2 illustrates the performance of the ensemble methods using the top four classifiers.

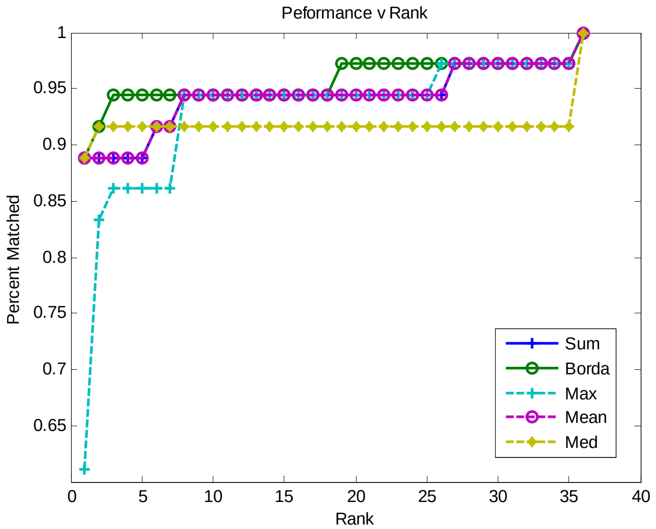


Figure 4.4-1: Ensemble performance of all Bands and PCTs.

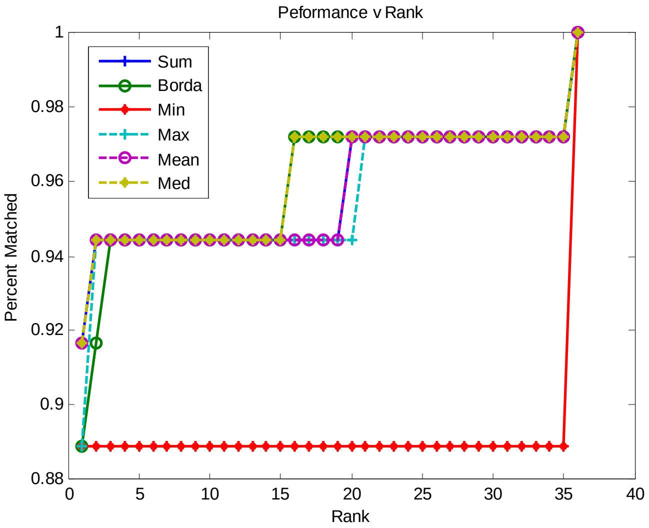


Figure 4.4-2: Ensemble performance of top 4 Bands and PCT (Band 2, 3, and 4, and 1st PCT).

It seems that the performances of the ensemble classifiers are lower than the best performing single classifier (1st PCT) where an accuracy of 97.22% is obtained at rank 4 by the 1st PCT versus at rank 16 by an ensemble of the top four using the Median rule. At first glance, this is somewhat counterintuitive since a combination of top performing classifiers should provide for an ensemble that is at least as good as its individual components, if not better; however, upon further analysis, the reason for the drop in performance may be due to the underachieving performance provided by Band 2, 3, and 4 in comparison to the 1st PCT which lessen the ensemble performance of the three classifiers. Figure 4.4-3 compares the best PCT and the best Uncorrelated Band performance with that of the best performing ensemble classifiers.

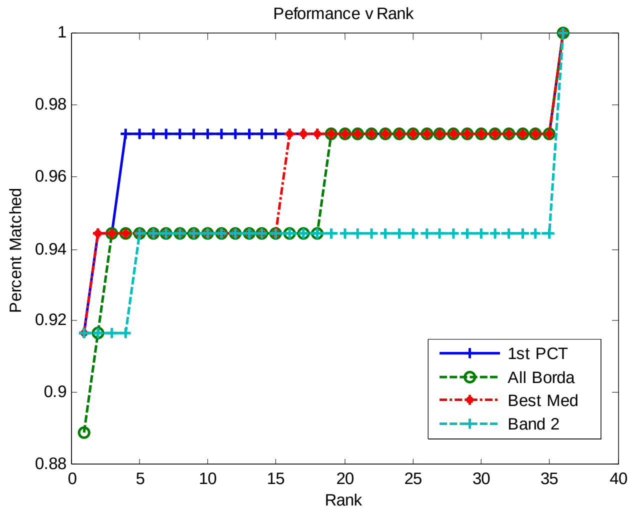
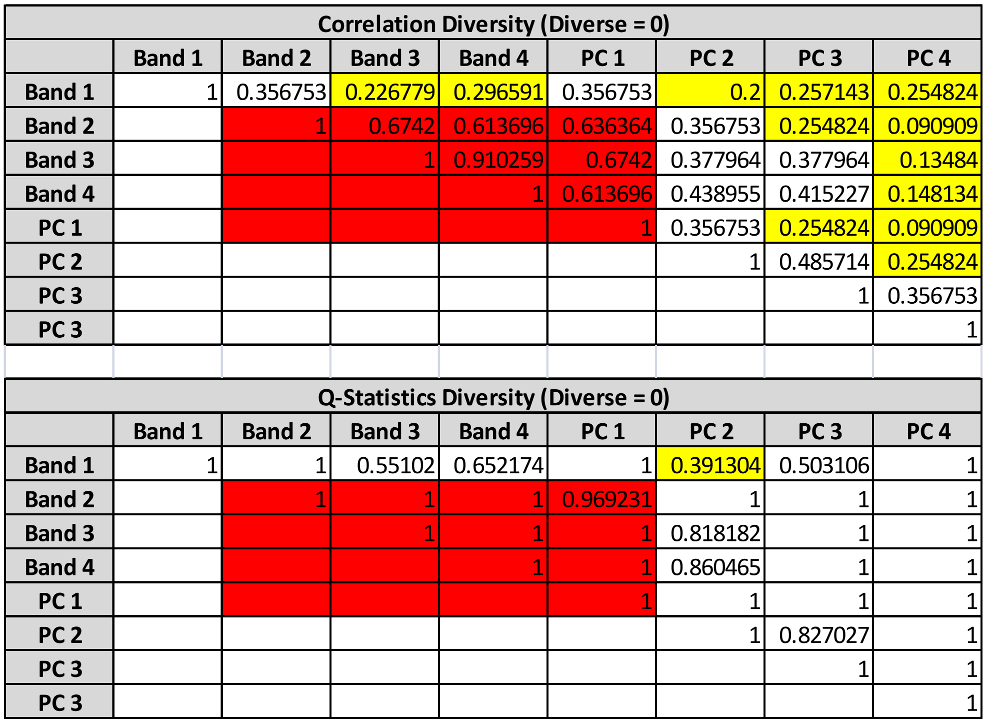


Figure 4.4-3: Performance comparison between 1st PCT, Borda count of all 8 classfiers, Median rule of top 4 classifiers, and Band 2.

One possible reason for the lack of improvement in the performance of the ensemble classifier is the lack of diversity between the various classifiers where they could all correctly classify or misclassify the same subject. Table 4.4-1 shows the Correlation Diversity matrix and the Q-Statistic Diversity matrix between each bands and PCTs which are obtained as described in 2.6. The red boxes mark the "sweet spot" of bands that have high accuracy whereas the yellow boxes highlight the lowest correlations and Q-statistic (high diversity) within each matrix. As we can see, all of the high performing bands are similar to other high performing bands and only have diversity to the low performing bands. This suggests that each of the high performing bands is sufficient by itself and could only gain more information by fusing with the lower performing bands, but as we have already shown above, ensembles of the lower performing bands can only result in a drop in the overall matching performance.

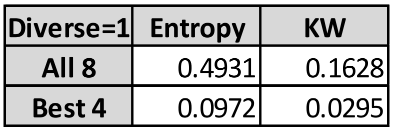
Table 4.4-1: Bands pair wise Correlation Diversity (top) and pair wise Q-Statistics Diversity (bottom).



Correlation Diversity (Diverse = 0)								
	Band 1	Band 2	Band 3	Band 4	PC 1	PC 2	PC 3	PC 4
Band 1	1	0.356753	0.226779	0.296591	0.356753	0.2	0.257143	0.254824
Band 2		1	0.6742	0.613696	0.636364	0.356753	0.254824	0.090909
Band 3		🟥	1	0.910259	0.6742	0.377964	0.377964	0.13484
Band 4		🟥	🟥	1	0.613696	0.438955	0.415227	0.148134
PC 1		🟥	🟥	🟥	1	0.356753	0.254824	0.090909
PC 2						1	0.485714	0.254824
PC 3							1	0.356753
PC 3								1

Q-Statistics Diversity (Diverse = 0)								
	Band 1	Band 2	Band 3	Band 4	PC 1	PC 2	PC 3	PC 4
Band 1	1	1	0.55102	0.652174	1	0.391304	0.503106	1
Band 2		1	1	1	0.969231	1	1	1
Band 3		🟥	1	1	1	0.818182	1	1
Band 4		🟥	🟥	1	1	0.860465	1	1
PC 1		🟥	🟥	🟥	1	1	1	1
PC 2						1	0.827027	1
PC 3							1	1
PC 3								1

Table 4.4-2: Bands overall diversity measures for all eight bands (All 8) and top four bands (Best 4).



Diverse=1	Entropy	KW
All 8	0.4931	0.1628
Best 4	0.0972	0.0295

Table 4.4-2 shows the overall diversity measure of all eight bands as a group and of the top four performing bands as a group which shows that the diversity are still fairly low with an entropy measure of 0.4931 and KW measure of 0.1628 despite the large variance of performance between all eight bands.

4.5. Performance of "Super-Optimum" Setting

It is apparent that the distortion of the test image of subject A07 is holding back the performance of the all of the classifiers causing the performance to stay below 100% until rank 36. This can be countered by increasing the size of Eraser to its high setting of

65 which allows for the inclusion of more pixels in the images which in turn allows for more possible SIFT descriptors and matching between the test and gallery image of subject A07. We therefore rerun the experiment on the Uncorrelated Bands using a "super-optimum" setting of low for Match = 3, Lag = 16, and Scales = 2, and high for Eraser = 65, Edge = 10 and Frame = 3. We also rerun the experiment on the PCT using a similar super-optimum setting of low for Match = 3 and Scales = 2, and high for Eraser = 65, Edge = 10 and Frame = 3 on the first four PCTs. Figure 4.5-1 shows the best performing band using the Super-Optimum setting in comparison to the top performing bands and ensemble in 4.4 and we can see that the matching performance can now approach 100% at a lower rank versus rank 36 which confirms our intuition.

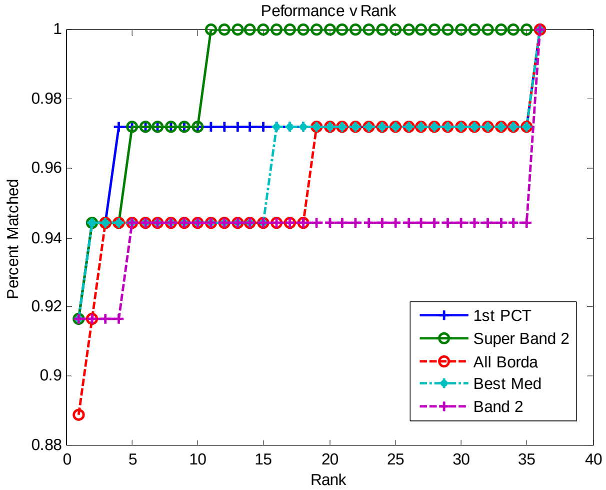


Figure 4.5-1: Performance comparison between Super-Optimum Band 2 with Figure 4.4-3.

The performance did not significantly improve when SIFT is applied to the 1st PCT using the Super-Optimum setting where the accuracy between rank 1 and rank 5 is lower than that of the Super-Optimum Band 2 although perfect matching is achieved much sooner with the Super-Optimum 1st PCT at rank 8. These results are illustrated in Figure 4.5-2 where they are also compared to the other better performing ensembles and, as expected, the ensemble performances are less than that of the 1st PCT due to the dominance of the 1st PCT performance. Figure 4.5-3 compares the three best performing classifiers which are the Optimum 1st PCT, Super-Optimum Band 2, and Super-Optimum 1st PCT with an ensemble of the Super-Optimum Band 2 and the Super-Optimum 1st PCT using Borda count. The Borda count ensemble provides an improvement in the lower rank at the cost of perfect matching at a higher rank of 13.

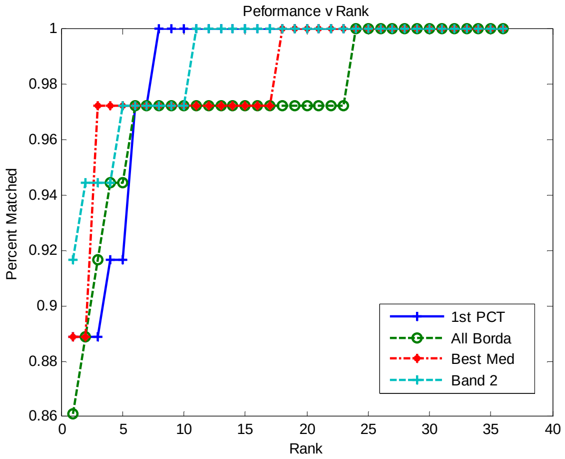


Figure 4.5-2: Performance comparison of the Super-Optimum setting between 1st PCT, Borda count of all 8 classfiers, Median rule of top 4 classifiers, and Band 2.

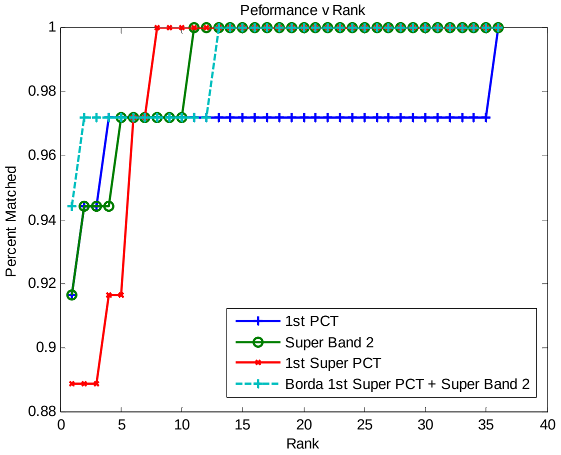


Figure 4.5-3: Comparison between Borda ensemble of 1st PCT and Band 2 of Super-Optimum with other top performing classifiers.

We then used the Super-Optimum settings for our algorithm and perform SIFT matching on the Band 2 and 1st PCT using images from sessions three, four and five as probes to study the robustness of our algorithm under temporal changes. As previously stated in 2.8, the images that we use were collected over the span of seven weeks. Although no time stamps were given for each particular session, we assume the time between each sessions are evenly spaced. Also, note that the number of subjects in each successive session is smaller than the previous one as indicated in 2.8. Based on Figure 4.5-4, it is shown that the ensemble performance varies only slightly between the test session and session three where perfect matching (28 subjects for session three) is obtained at rank 16 while a matching of 96% is obtained at a low rank of 2. However, a stronger support for temporal robustness is given by the performance of session 4 (22 subjects) where perfect matching is obtained at rank 1 using the ensemble method as shown in Figure 4.5-5.

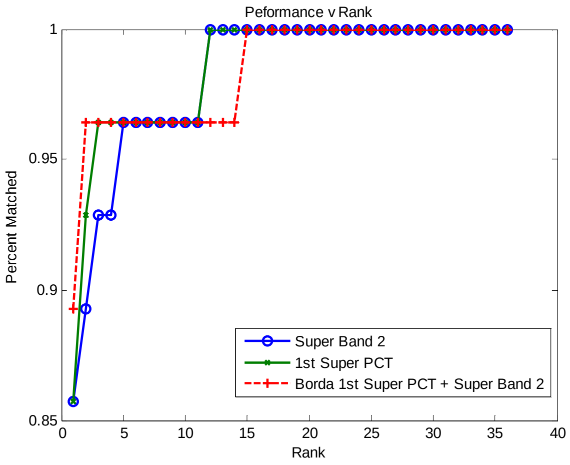


Figure 4.5-4: Matching performance of Session 3 images using the final settings.

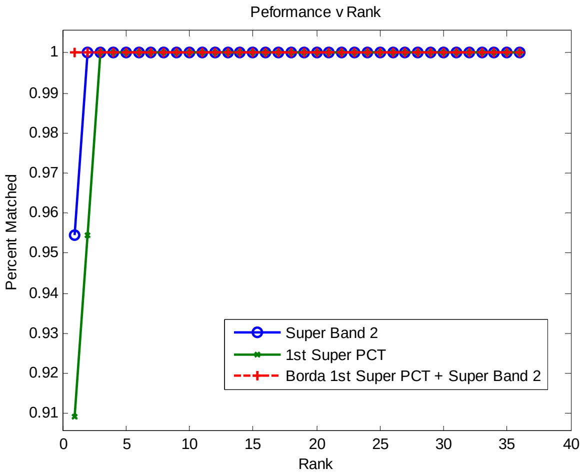


Figure 4.5-5: Matching performance of Session 4 images using the final settings.

The performance appears to have degraded somewhat significantly for session five (16 subjects) based on Figure 4.5-6. However, this is due to the smaller number of subjects in the probe set because only three subjects were not able to be matched at rank 1 and perfect matching is obtained at rank 18 which is comparable to the performance of session three.

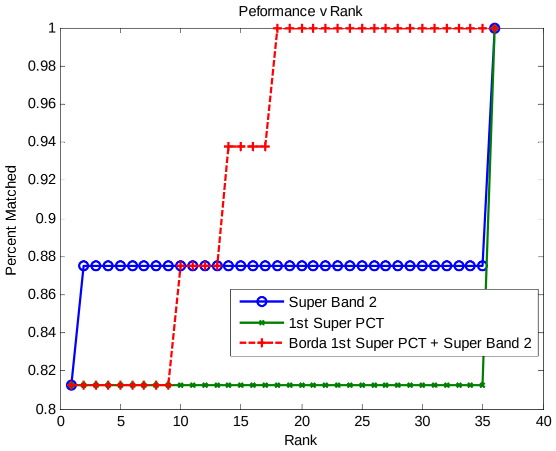


Figure 4.5-6: Matching performance of Session 5 images using the final settings.

In comparison to the results obtained by Ryer et al. (2011), our method appears to perform better based on the additional sets of data of session 3, 4, and 5. Figure 4.5-7, 8,

& 9 illustrate the overall performance that they obtained from their methodology. Unlike our implementation, SIFT is not applied to the full spectral range of the images but their SIFT matching extends to include the hair component of the subjects (hf-sift) whereas we only considered the skin component. For session 3, we are able to achieve above 95% matching at rank 2 as opposed to rank 6 in their implementation. We are also able to outperform their implementation in session 4 where we achieve perfect matching at rank 1 versus rank 2 using hf-sift. However, the result from session 5 clearly shows that applying SIFT to the full spectral range greatly improves the matching performance where we achieve over 80% matching at rank 1 and 100% matching at rank 18 in contrast to 21% matching at rank 1 and 100% at rank 34 provided by hf-sift.

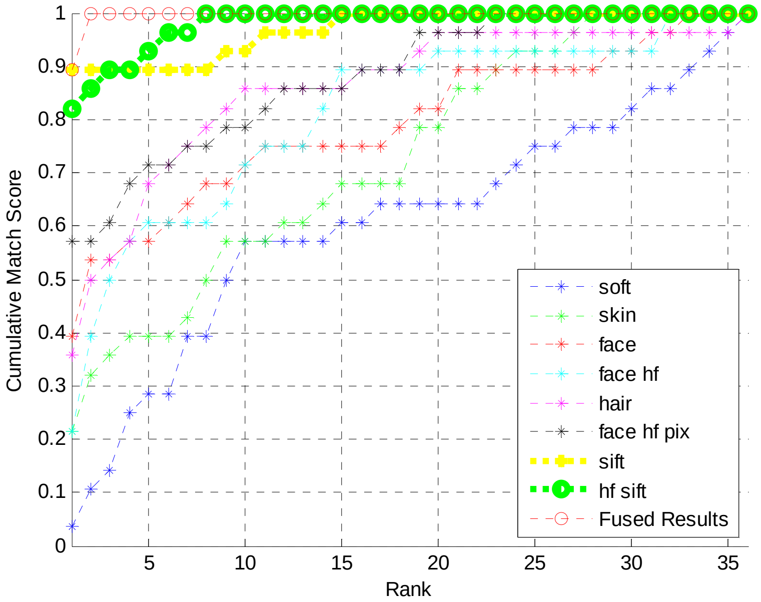


Figure 4.5-7: Ryer et al. session 3 performances.

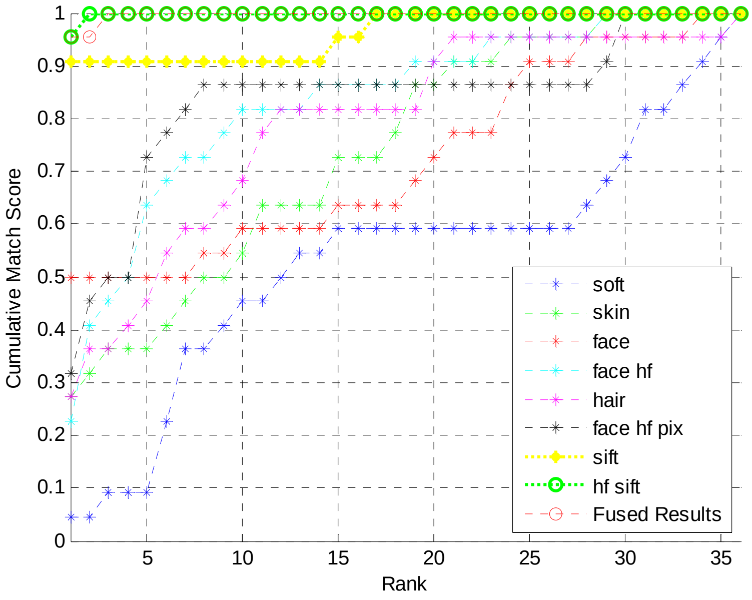


Figure 4.5-8: Ryer et al. session 4 performances.

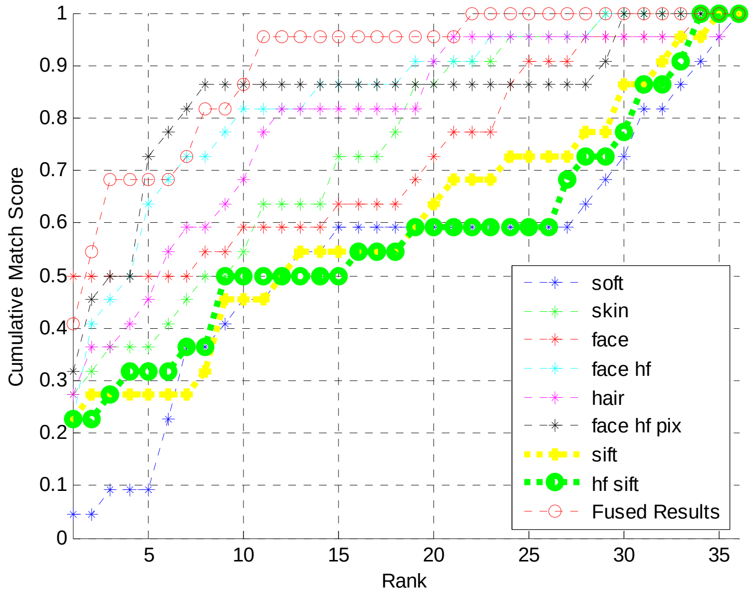


Figure 4.5-9: Ryer et al. session 5 performances.

5. Discussion

5.1. Conclusion

Correlation between the bands of the hyperspectral images is first studied to see if the spectral dimension can be reduced without too much loss of information. This is followed by running a designed experiment by treating six of the parameters within the algorithms as factors. The purpose for running the designed experiment is two-fold; the first is to see whether or not the parameters are significant with respect to the matching performance, and the second is to obtain an optimum setting for the algorithm that maximizes the expected performance.

We then apply a SIFT matching on the PCTs using the optimum setting and found that performance is best within the first 5 ranks. An ensemble matching is then performed by combining the results of the individual bands and PCTs to see if better performance is possible. The performance of fusing all bands and PCTs is lower than that of fusing the top four performing bands and the first PCT. However, the performance of the 1st PCT by itself is still better than all of the other individual and ensemble performance. Also, perfect matching could not be attained until rank 36 due to the artifacts present within the image of subject A08 caused by the Eraser Size of the NDSI method.

An adjustment of the parameter is then made to resolve the problem by using its high setting which allow for more skin pixels to be included. This improves the matching performance where a matching of 94.44% is attained at rank 1 and perfect matching is obtained at rank 13 by an ensemble of Band 2 and 1st PCT using the Borda Count rule.

We then show that the algorithm is robust under temporal changes and that the performance does not vary significantly for session three, four and five where matching performance at rank 1 are 89%, 100%, and 81%, respectively, and perfect matching is obtained at rank 16, 1, and 18, respectively.

5.2. Thesis Contribution

Our research has shown that an application of SIFT on hyperspectral images for face recognition improves the overall matching accuracy in comparison to a SIFT matching of a single spectral band as conducted by Ryer et al. (2011). In particular, an ensemble SIFT matching using the Borda Rule of the first PCT and a band with a spectral range of 660-670nm provides a strong face recognition algorithm. We have also shown that our method is somewhat robust under temporal changes since all five sessions were collected over the span of two months. Figure 5.2-1 illustrates the proposal by Ryer et al. of an incremental approach to performing fusion hierarchy for hyperspectral face images. Our research essentially combined the last two steps of this hierarchy into a single Spectral-Spatial Recognition process as illustrated in Figure 5.2-2. We believe that the strong result obtained by this achievement shows that face recognition in the realm of hyperspectral is promising and can further advance the fidelity of face recognition.

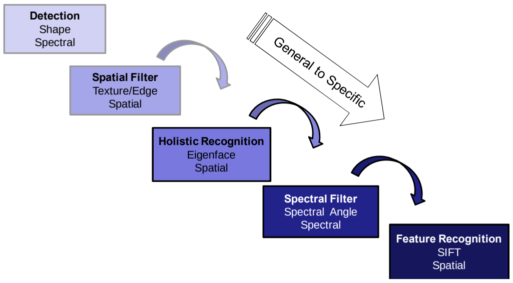


Figure 5.2-1: Hyperspectral Face Recognition Fusion Hierarchy (Ryer, Bihl, Bauer, & Rogers, 2011)

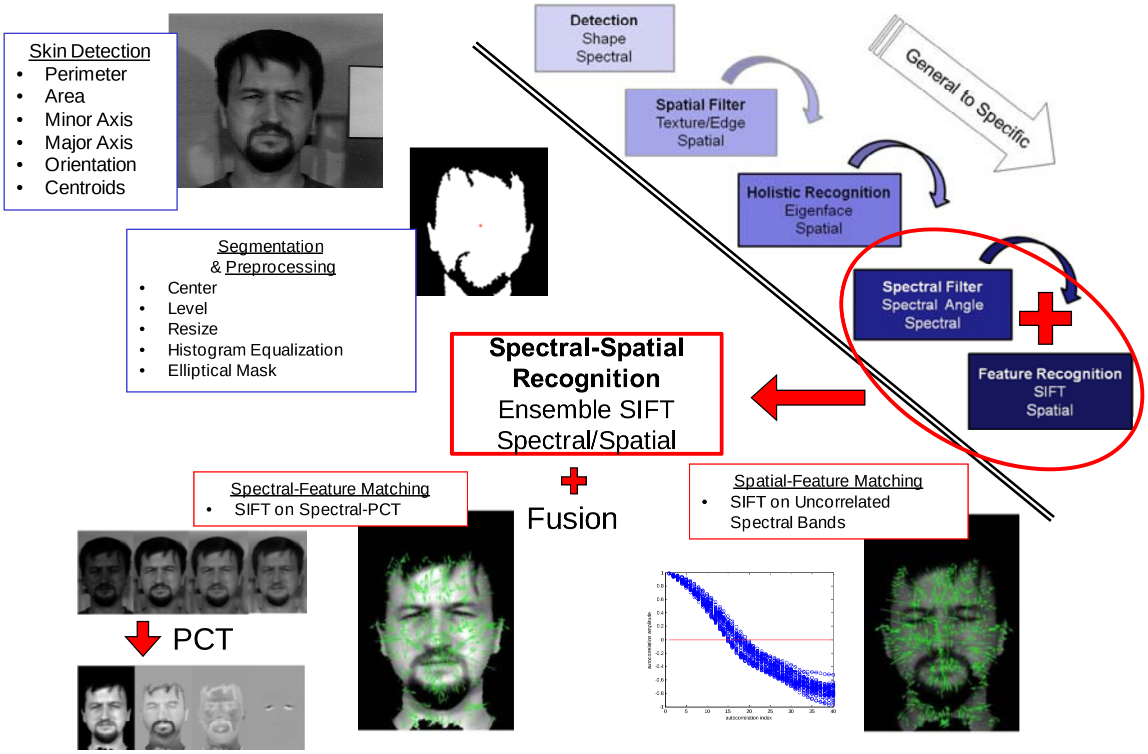


Figure 5.2-2: Thesis contribution chart (contribution outlined in red).

5.3. Issues Encountered

The main disadvantage of our method is that, although SIFT provides a robust feature detection, the implementation by Vedaldi (2006) that we utilize is somewhat computationally intensive even for a single band matching of the database as previously acknowledge by Ryer (2011). In our application, the runtime for matching four uncorrelated bands of a pair of subjects was around 30 seconds using a Hewlett-Packard DC5850 Microtower desktop with an AMD Athlon 64 X2 7750 processor and 4GB PC2-6400 RAM. This amounts to a total runtime of 10 hours for matching every subject in a database of 36 subjects. One possible solution is to perform parallel computing where the runtime is then decreased by a factor of total number of machines.

Performance was also downgraded due to the artifacts (e.g. movements and jitter between the bands, noise in the lower and higher bands etc.) present within the images where applications of the NDSI method were not able to fully capture the skin component of a few subjects which in turn hinders the SIFT algorithm to locate all of the descriptors within the subjects' face. However, we do not have control over the cleanliness or fidelity of the data since it was obtained from another party, and a cleaner set of data could have provided a better performance, but the silver lining is that "dirtier" data provided a better representation of the type of data that might be collected in an uncontrolled environment as opposed to a lab.

5.4. Future Research

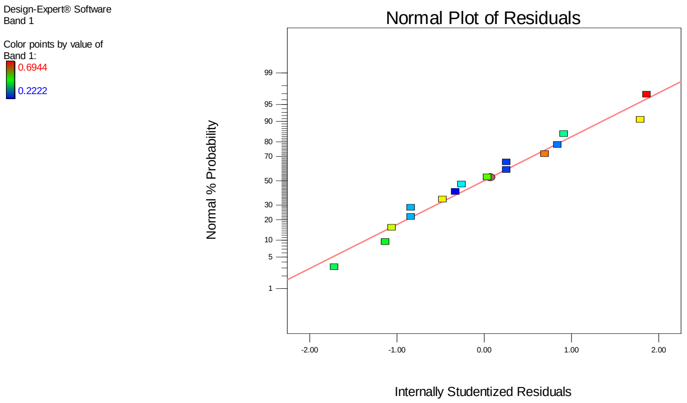
Future research that could contribute to the field of face recognition and improve the results obtained by the author includes:

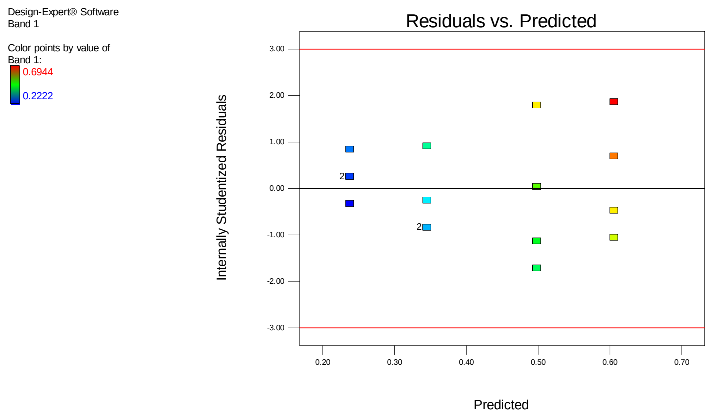
- Investigating the performance of pose variation using the same methodology. This will require the collection of a new set of data that contains pose variation.
- Performing an expanded Design of Experiment since it is shown that the range for the NDSI setting does not fully capture the skin component of a few subjects.
- Investigating the performance of negatively correlated bands which might provide the diversity needed for an improved ensemble performance.
- Creating an adaptive algorithm with a dynamic library that assimilates an out of library target.
- Applying QUEST (Ryer, Bihl, Bauer, & Rogers, 2011) methodology to current work.

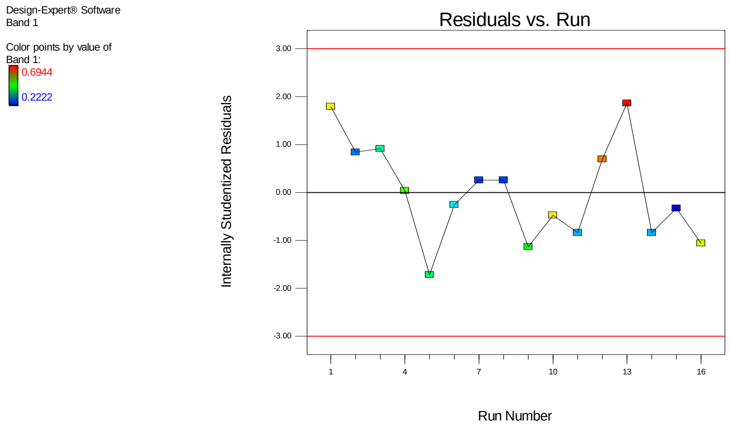
Appendix A: Designed Experiment

Order		Factors						Response (Proportion Matched)			
Standard Order	Run Order	Eraser	SIFT	Lag	Scales	Edge	Frames	Band 1	Band 2	Band 3	Band 4
2	1	65	3	16	2	10	2	0.5833	0.8611	0.8056	0.7778
12	2	65	7	16	3	5	2	0.2778	0.7778	0.7222	0.5833
16	3	65	7	18	3	10	3	0.3889	0.8333	0.75	0.6111
1	4	5	3	16	2	5	2	0.5	0.8611	0.75	0.6944
14	5	65	3	18	3	5	2	0.4167	0.8611	0.7222	0.6944
15	6	5	7	18	3	5	3	0.3333	0.8611	0.6944	0.6667
8	7	65	7	18	2	10	2	0.25	0.7778	0.7222	0.5833
7	8	5	7	18	2	5	2	0.25	0.75	0.6389	0.6389
13	9	5	3	18	3	10	2	0.4444	0.8889	0.8333	0.7222
5	10	5	3	18	2	10	3	0.5833	0.9167	0.8333	0.8611
4	11	65	7	16	2	5	3	0.3056	0.8889	0.75	0.7222
6	12	65	3	18	2	5	3	0.6389	0.9167	0.8056	0.75
10	13	65	3	16	3	10	3	0.6944	0.8611	0.8056	0.8056
3	14	5	7	16	2	10	3	0.3056	0.8889	0.8333	0.7222
11	15	5	7	16	3	10	2	0.2222	0.8889	0.7778	0.6744
9	16	5	3	16	3	5	3	0.5556	0.8611	0.8056	0.8056

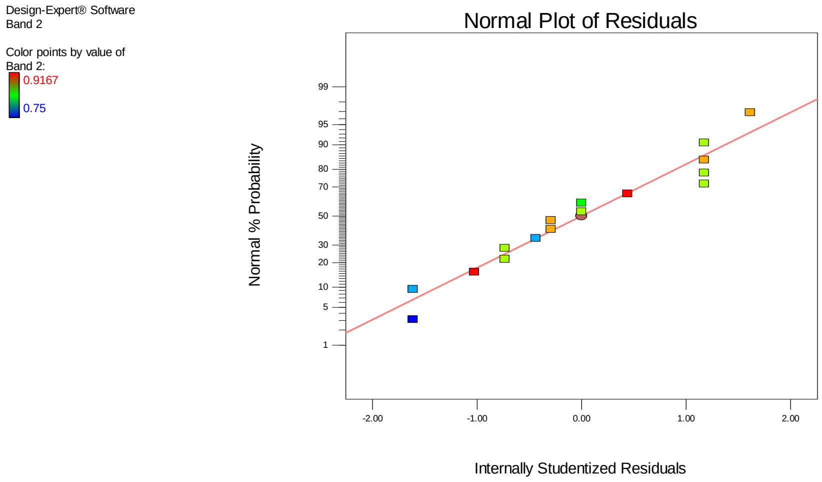
Band 1						
ANOVA for selected factorial model						
Analysis of variance table [Partial sum of squares - Type III]						
	Sum of		Mean	F	p-value	
Source	Squares	df	Square	Value	Prob > F	
Model	0.317587	2	0.158793	57.18086	< 0.0001	significant
B-SIFT	0.271233	1	0.271233	97.6698	< 0.0001	
F-Frame	0.046354	1	0.046354	16.69192	0.0013	
Residual	0.036101	13	0.002777			
Cor Total	0.353688	15				

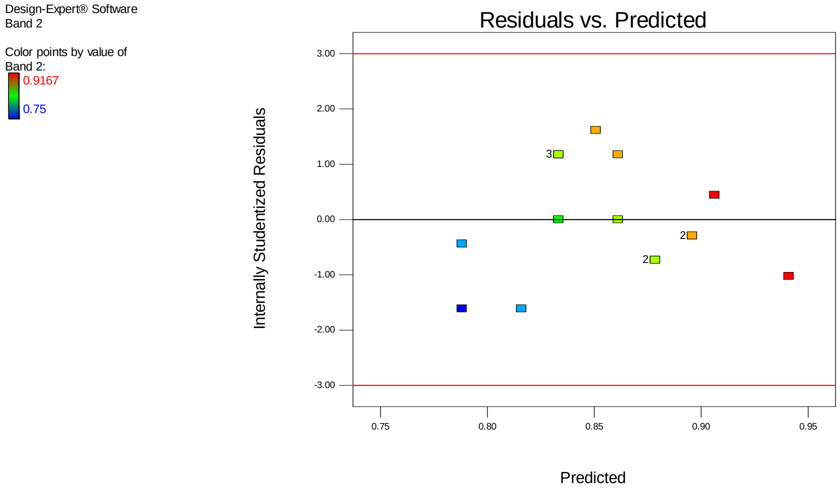


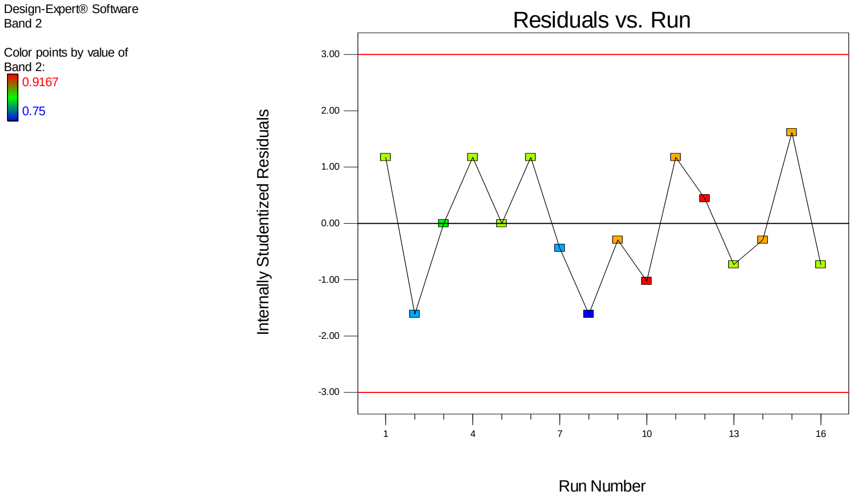




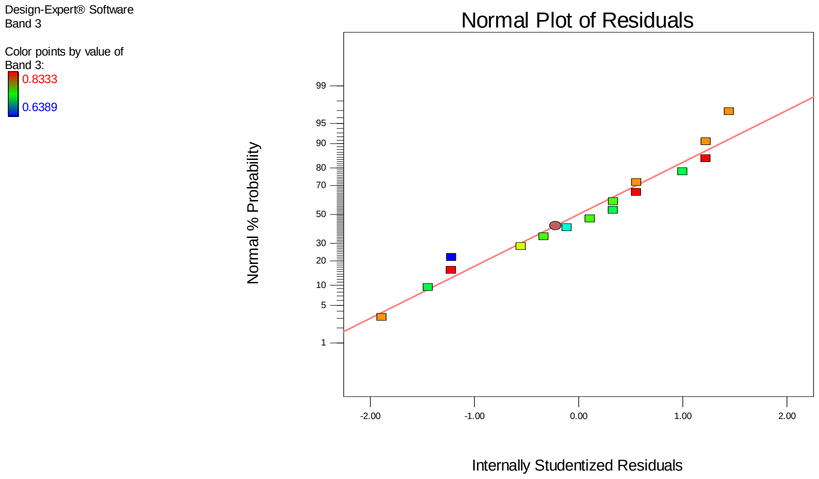
Band 2						
ANOVA for selected factorial model						
Analysis of variance table [Partial sum of squares - Type III]						
	Sum of		Mean	F	p-value	
Source	Squares	df	Square	Value	Prob > F	
Model	0.026869	5	0.005374	5.99302	0.0081	significant
A-Eraser	0.001206	1	0.001206	1.344752	0.2731	
B-SIFT	0.00815	1	0.00815	9.088508	0.0130	
E-Edge	0.001206	1	0.001206	1.344752	0.2731	
F-Frame	0.00815	1	0.00815	9.088508	0.0130	
AE	0.008159	1	0.008159	9.098579	0.0130	
Residual	0.008967	10	0.000897			
Cor Total	0.035836	15				

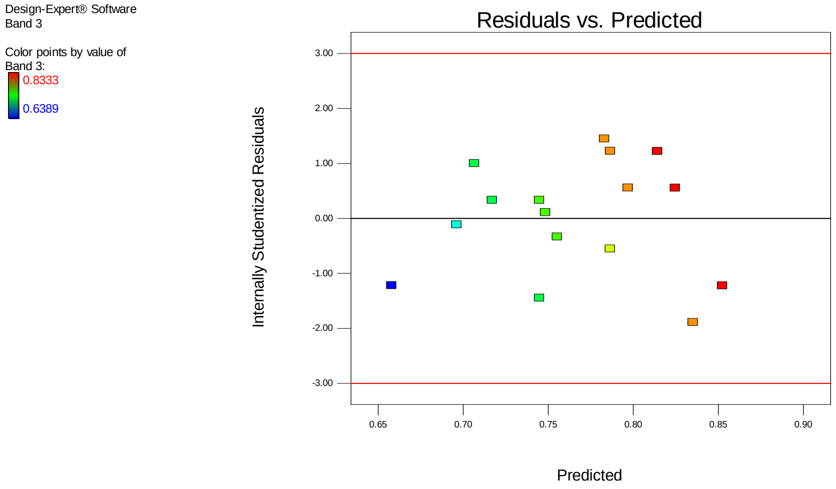

Design-Expert® Software
Band 2
Color points by value of
Band 2:
0.9167
0.75
Normal Plot of Residuals
Normal % Probability
99
95
90
80
70
50
30
20
10
5
1
-2.00
-1.00
0.00
1.00
2.00
Internally Studentized Residuals

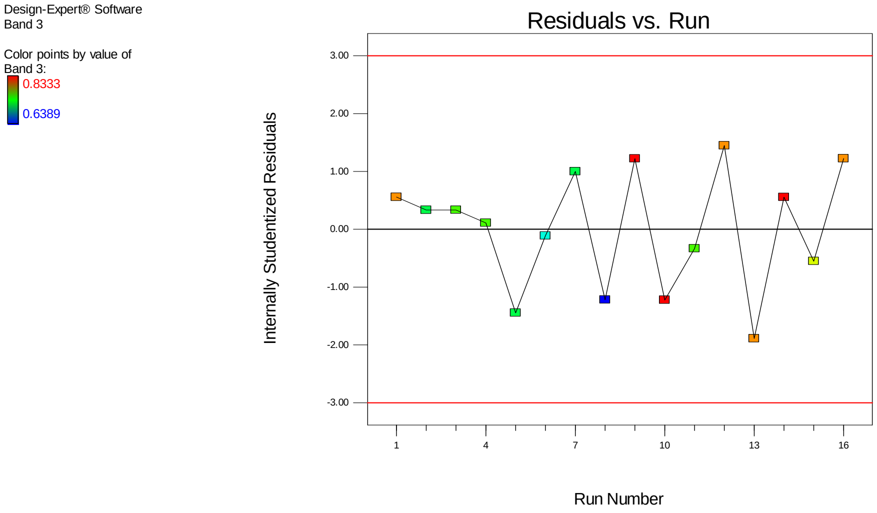

Design-Expert® Software
Band 2
Color points by value of
Band 2:
0.9167
0.75
Residuals vs. Predicted
Internally Studentized Residuals
3.00
2.00
1.00
0.00
-1.00
-2.00
-3.00
3
2
2
0.75
0.80
0.85
0.90
0.95
Predicted


Design-Expert® Software
Band 2
Color points by value of
Band 2:
0.9167
0.75
Residuals vs. Run
Internally Studentized Residuals
3.00
2.00
1.00
0.00
-1.00
-2.00
-3.00
1
4
7
10
13
16
Run Number

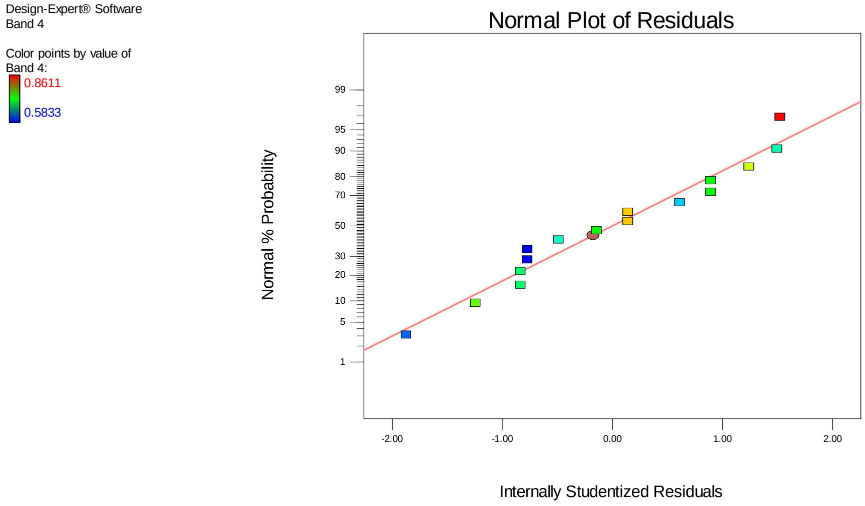
Band 3						
ANOVA for selected factorial model						
Analysis of variance table [Partial sum of squares - Type III]						
	Sum of		Mean	F	p-value	
Source	Squares	df	Square	Value	Prob > F	
Model	0.043895	6	0.007316	16.85259	0.0002	significant
A-Eraser	0.000433	1	0.000433	0.996623	0.3442	
B-SIFT	0.013948	1	0.013948	32.12952	0.0003	
C-Lag	0.003913	1	0.003913	9.012785	0.0149	
E-Edge	0.013936	1	0.013936	32.10232	0.0003	
F-Frame	0.005837	1	0.005837	13.44594	0.0052	
AE	0.005829	1	0.005829	13.42834	0.0052	
Residual	0.003907	9	0.000434			
Cor Total	0.047802	15				

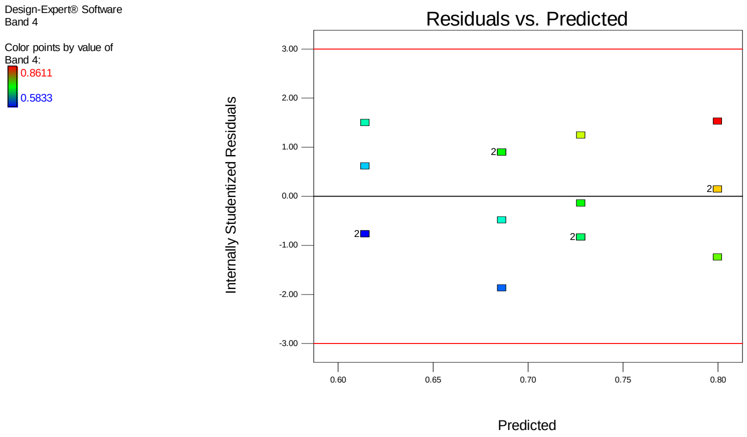


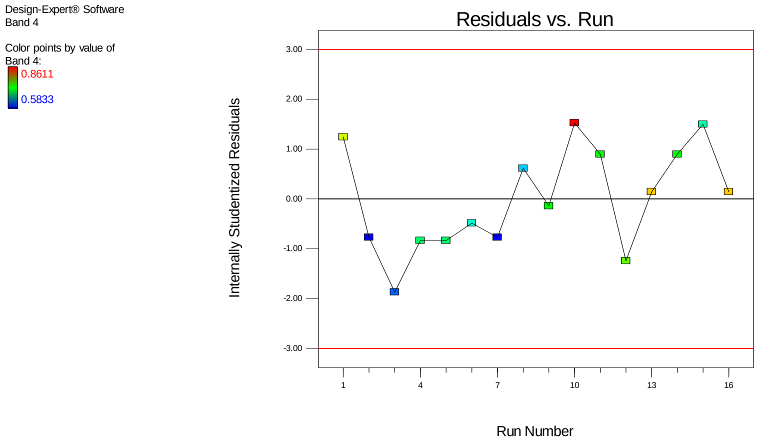




Band 4						
ANOVA for selected factorial model						
Analysis of variance table [Partial sum of squares - Type III]						
	Sum of		Mean	F	p-value	
Source	Squares	df	Square	Value	Prob > F	
Model	0.072364	2	0.036182	18.21476	0.0002	significant
B-SIFT	0.051643	1	0.051643	25.99786	0.0002	
F-Frame	0.020722	1	0.020722	10.43165	0.0066	
Residual	0.025823	13	0.001986			
Cor Total	0.098188	15				




Design-Expert® Software
Band 4
Color points by value of
Band 4:
0.8611
0.5833
Residuals vs. Predicted
Internally Studentized Residuals
3.00
2.00
1.00
0.00
-1.00
-2.00
-3.00
2
2
2
2
0.60
0.65
0.70
0.75
0.80
Predicted


Design-Expert® Software
Band 4
Color points by value of
Band 4:
0.8611
0.5833
Residuals vs. Run
Internally Studentized Residuals
3.00
2.00
1.00
0.00
-1.00
-2.00
-3.00
1
4
7
10
13
16
Run Number

Appendix B: MatLab® Code

1)

```
function [poll,matches,probe,target] = hsi_sift_match(...

probe_data_cube,target_data_cube,eraser_size,sift_threshold,interval)

%  hsi_sift_match matches the SIFT descriptors of the bands within the
%  hyperspectral image of the probe and target and takes in
%  probe_data_cube, target_data_cube, eraser_size, sift_threshold, and
%  interval as input variables and gives out poll, matches, probe, and
%  target as output variables.
%
%  Input:
%  probe_data_cube - a hyperspectral datacube of a probe
%  target_data_cube - a hyperspectral datacube of a target
%  eraser_size - eraser_size value called by poorman_skin_detection.m
%  sift_threshold - matching threshold called by siftmatch.m
%  interval - lag between bands to be matched
%
%  Output:
%  poll - a 1xPC matrix where each column is the number of possible
matching
%  of each band given by siftmatch.m
%  matches - a structure array that lists the frames of the matching
%  descriptors of each band
%  probe - a structure array containing all of the sift.m outputs for
the
%  probe
%  target - a structure array containing all of the sift.m outputs for
the
%  target

tic
disp('Detecting Faces')
eraser_size = eraser_size;
probe_cube = probe_data_cube;
target_cube = target_data_cube;
skin_probe = uint8(poorman_skin_detection(probe_cube,eraser_size));
skin_target = uint8(poorman_skin_detection(target_cube,eraser_size));
toc

tic
disp('Grabbing Faces New')
size1 = size(probe_cube,1);
size2 = size(probe_cube,2);
size3 = size(probe_cube,3);
probe_data = zeros(size1,size2,size3);
target_data = zeros(size1,size2,size3);
for index_band = 1:size(probe_cube,3)
    skin_probe = reshape(skin_probe,size1*size2,1);
    probe_cube2 = reshape(probe_cube,...
        size1*size2,size3);
    probe_data2 = skin_probe.*probe_cube2(:,index_band);
     probe_data(:,:,index_band) = reshape(probe_data2,size1,size2);
end

for index_band = 1:size(target_cube,3)
    skin_target = reshape(skin_target,size1*size2,1);
    target_cube2 = reshape(target_cube,...
       size1*size2,size3);
    target_data2 = skin_target.*target_cube2(:,index_band);
    target_data(:,:,index_band) = reshape(target_data2,size1,size2);
end
toc

lag = interval;

tic
disp('Applying SIFT to Bands')
k=1;
for index_band = 6:lag:size(probe_data,3)
    probe.face{k} = probe_data(:,:,index_band);
    [probe.frames{k},probe.descriptors{k}] = ...
        sift(probe_data(:,:,index_band));
    probe.band{k} = index_band;
    target.face{k} = target_data(:,:,index_band);
    [target.frames{k},target.descriptors{k}] = ...
        sift(target_data(:,:,index_band));
    target.band{k} = index_band;
    k = k+1;
end
toc

tic
disp('Matching Probe and Target')
matching_bands = max(size(probe.band,2),size(target.band,2));

k=1;
matching_thresh = sift_threshold;
for index_band = 1:matching_bands
    matches.match{k} = siftmatch(probe.descriptors{index_band},...
        target.descriptors{index_band},matching_thresh);
    matches.bands{k} = [probe.band{index_band};...
        target.band{index_band}];
    k=k+1;
end
toc

poll = zeros(1,size(matches.match,2));
for index_matches = 1:size(matches.match,2)
    poll(index_matches) = size(matches.match{index_matches},2);
end
```

2)

```
function [eigface,LplotT] = eigenface(data_cube,eraser_size)

%  eigenface detects the skin component of a hyperspectral datacube and
%  computes the principle components of the skin and takes in takes in
%  data_cube and eraser_size as inputs and gives out eigface and LplotT
as
%  outputs
%
%  Input:
%  data_cube - a hyperspectral datacube of a subject
%  eraser_size - the eraser_size value called by
poorman_skin_detection.m
%
%  Output:
%  eigface - a principle components datacube of the subject
%  LplotT - the eigenvalues associated with each principle components

tic
disp('Detecting Faces')
eraser_size = eraser_size;
target_cube = data_cube;
skin_target = uint8(poorman_skin_detection(target_cube,eraser_size));
clear data_cube
toc

tic
disp('Grabbing Faces New')
size1 = size(target_cube,1);
size2 = size(target_cube,2);
size3 = size(target_cube,3);
target_data = zeros(size1,size2,size3);
skin_target = reshape(skin_target,size1*size2,1);
target_cube2 = reshape(target_cube,size1*size2,size3);
for index_band = 1:size(target_cube,3)
    target_data2 = skin_target.*target_cube2(:,index_band);
    target_data(:,:,index_band) = reshape(target_data2,size1,size2);
end

clear skin_target skin_gallery target_cube target_cube2 target_data2
toc

tic
disp('Finding Eigenfaces')
dim_adjustment = 0;

size1 = size(target_data,1);
size2 = size(target_data,2);
size3 = size(target_data,3);

target_matrix = reshape(target_data,size1*size2,size3);
clear data_cube
```

```
good_bands = [6:60];

good_target_matrix = double(target_matrix(:,good_bands));
clear target_matrix


dims = size(good_target_matrix,2);

%------------------------------Perform PCA-----------------------------
----
[AcT,LcT,TotVarCompCT,YscorCT]=Center_and_PCA_optimized(good_target_mat
rix);
LplotT=diag(LcT);
%checks for eigenvalues 10^-4 and smaller and moves the endpoint of the
%eigenvalue curve to the point where eigenvalues are greater than 10^-4
%so that the MDSL method in the next section is not biased by
pathological
%cases where the endpoint of the log scale eigenvalue curve has
extremely
%small endpoints and grossly alters the theoretical shape of the curve
that
%should arise for eigenvalues of covariance matrices of spectral data
%that follow the LMM
while LplotT(dims)<=10^-4;
    dims=dims-1;
end
LT=log10(LplotT(1:dims));

%----------------------------------------------------------------------
----
%---------------------Dimensionality Assessment------------------------
----
%slope of line connecting endpoints of scree plot of eigenvalues
m_slopeT = (LT(1)- LT(dims))/(1-dims);

%calculate Euclidean distances from scree plot curve to line connecting
%endpoints
dummy = ones(dims,1);
x_int = (LT - LT(1)*dummy + m_slopeT*dummy + (1:dims)'./m_slopeT)./...
    (m_slopeT + 1/m_slopeT) ;
y_int = LT(1)*dummy + m_slopeT.*(x_int - dummy);
Eqdist = sqrt( ( (1:dims)' - x_int).^2 + (LT - y_int).^2)' ;
clear x_int y_int dummy m_slopeT
%find the point on the log scale eigencurve curve with the largest
distance
%from the line connecting the endpoints
[max_EqdistT, index_dimT] = max(Eqdist);
clear Eqdist
reduced_dimT = index_dimT;
kT=reduced_dimT-1;
kT=kT+dim_adjustment;
percent_varT=TotVarCompCT(kT,1);
YT=YscorCT(:,1:kT);
```

```
clear YscorCT;

eigface = reshape(YT,size1,size2,kT);

toc
```

3)

```
function [poll,matches,probe,target] = hsi_sift_match_v3(...
    probe_data_cube,target_data_cube,sift_threshold,PC)

%  hsi_sift_match_v3 matches the SIFT descriptors of the eigenfaces of
%  a probe and a target and takes in probe_data_cube, target_data_cube,
%  sift_threshold, and PC as input variables and gives out poll,
matches,
%  probe, and target as output variables.
%
%  Input:
%  probe_data_cube - a set of eigenfaces of a probe obtained using
%  eigenface.m
%  target_data_cube - a set of eigenfaces of a target obtained using
%  eigenface.m
%  sift_threshold - matching threshold called by siftmatch.m
%  PC - number of eigenfaces matching
%
%  Output:
%  poll - a 1xPC matrix where each column is the number of possible
matching
%  of each eigenfaces given by siftmatch.m
%  matches - a structure array that lists the frames of the matching
%  descriptors of each eigenface
%  probe - a structure array containing all of the sift.m outputs for
the
%  probe
%  target - a structure array containing all of the sift.m outputs for
the
%  target

tic
principal_comp = PC;

probe_data = probe_data_cube;
target_data = target_data_cube;

disp('Applying SIFT to Eigenfaces')
k=1;
for index_band = 1:principal_comp
    probe.face{k} = probe_data(:,:,index_band);
    [probe.frames{k},probe.descriptors{k}] = ...
        sift2(probe_data(:,:,index_band));
    probe.band{k} = index_band;
    target.face{k} = target_data(:,:,index_band);
    [target.frames{k},target.descriptors{k}] = ...
        sift2(target_data(:,:,index_band));
    target.band{k} = index_band;
    k = k+1;
end
toc

tic
```

```
disp('Matching Probe and Target')
matching_bands = max(size(probe.band,2),size(target.band,2));

k=1;
matching_thresh = sift_threshold;
for index_band = 1:matching_bands
    matches.match{k} = siftmatch(probe.descriptors{index_band},...
        target.descriptors{index_band},matching_thresh);
    matches.bands{k} = [probe.band{index_band};...
        target.band{index_band}];
    k=k+1;
end
toc

poll = zeros(1,size(matches.match,2));

for index_matches = 1:size(matches.match,2)
    poll(index_matches) = size(matches.match{index_matches},2);
end
```

4)

```
function [rankmat,sorted_rank] = rank_evaluator(scores)

%  rank_evaluator calculates the the rank at which each subject is
matched.
%  It takes in a 3 dimensional matrix of matching scores as input and
gives
%  a matrix of matching ranks for each pair of probe and target called
%  rankmat and a matrix of ranks at which each probe is correctly
matched
%  called sorted_rank as output
%
%  Input:
%  scores - a 3 dimensional matrix of matching scores
%
%  Output:
%  rankmat - a 3 dimensional matrix where the rows are probes and the
%  number of columns is equal to the number of target, rankmat(:,:,1)
is
%  the target that is matched to the probe, rankmat(:,:,2) is the
matching
%  score, rankmat(:,:,3) is the rank of each matching, and
rankmat(:,:,4)
%  is a binary indicator for a correct matching
%  sorted_rank - a matrix where the rows are the probes and columns are
the
%  ranks and each cell is a binary indicator of a matching

temp = scores;
rankmat = zeros(36,36,4);
%  Sort matching based on matching scores from highest to lowest
for index_target = 1:36
    for index_gallery = 1:36
        gallery = find(temp(index_target,:)==...
            max(temp(index_target,:)),1);
        rankmat(index_target,index_gallery,1) = gallery;
        rankmat(index_target,index_gallery,2) =
max(temp(index_target,:));
        temp(index_target,gallery) = -10^3;
    end
end

%  Find the rank for each matching and whether or not it is a correct
%  matching.
for index_target = 1:36
    for index_gallery = 1:36
        if index_target == rankmat(index_target,index_gallery,1)
            rankmat(index_target,index_gallery,3) = index_gallery;
            rankmat(index_target,index_gallery,4) = 1;
        else
            rankmat(index_target,index_gallery,3) = index_gallery;
            rankmat(index_target,index_gallery,4) = 0;
        end
    end
end

%  Any matching that has a score of 0 is penalized and given the
%  lowest rank.
for index_target = 1:36
    for index_gallery = 2:36
        if rankmat(index_target,index_gallery,2) ==...
                rankmat(index_target,index_gallery-1,2) && ...
                rankmat(index_target,index_gallery-1,3) < ...
                rankmat(index_target,index_gallery,3);
            rankmat(index_target,index_gallery,3) =...
                rankmat(index_target,index_gallery-1,3);
        end
        if rankmat(index_target,index_gallery,2) == 0
            rankmat(index_target,index_gallery,3) = ...
                36;
        end
    end
end
% rankmat(:,1,3) = 1;

%  Sort the correct matchings based on the rank for each matching
sorted_rank = zeros(36,36);
for index_target = 1:36
    gallery = find(rankmat(index_target,:,4)==1,1);
    rank_index = rankmat(index_target,gallery,3);
    sorted_rank(index_target,rank_index) = 1;
end
```

5)

```
function [] =
rank_performance_plot(performancemat,linespec,color,linewidth)

%  rank_performance_plot calculates and plots the matching performance
of
%  an algorithm based on the rank.  It takes in a matrix of performance
%  given by rank_evaluator.m
%
%  Input:
%  performancemat - a performance matrix given by rank_evaluator.m
%  linespec - type of lines for plotting
%  color - color of lines
%  linewidth - size of lines
%
%  Output:
%  a plot of matching performance

if nargin == 1;
    color = 'b';
end

%  Sums up the number of matched subject by rank to calculate
proportion
%  matched
performance = zeros(1,36);
for index_rank = 1:36
    performance(1,index_rank) =...
        sum(sum(performancemat(:,1:index_rank),2),1)/36;
end

plot(performance,linespec,'color',color,'linewidth',linewidth)
xlabel('Rank')
ylabel('Percent Matched')
title('Peformance v Rank')
```

All codes published in MATLAB® 7.10

Appendix C: Blue Dart

Replace DNA matching with face recognition?

Face recognition is a useful and valuable biometric due to its non-invasive nature. Face recognition is a well established field of research dating back to the late 1980s. The most obvious application for face recognition is for real-time target detection in a crowded and high flowing environment. However, another useful application is for person identification that does not require real-time capability where the facial feature of the target has been altered due to aging or physical trauma. CNN reported on May 2nd 2011 that one of the methods of identifying Osama bin Laden after his body was captured was with the use of a face recognition method. This suggests that if the method being used can provide high fidelity regardless of its computing time, then it is highly desirable. This could be very useful when other biometrics in indentifying the target that would certainly perform well under such condition, such as DNA matching, is not available. However, this requires that the face recognition algorithm to be highly robust with respect to the temporal and spatial changes.

Face recognition through hyperspectral images is a concept which is still in its infancy. Although the conventional method of face recognition using Red-Green-Blue (RGB) or grayscale images has been advanced over the last twenty years, these methods are still shown to have weak performance whenever there are variations or changes in lighting, pose, or temporal aspect of the subjects. A hyperspectral representation of an image captures more information that is available within a scene than a RGB image therefore it is beneficial to study the performance of face recognition using a hyperspectral representation of the subjects' faces.

Although a hyperspectral image gives us more data to work with, the overall data can still be highly correlated. It is therefore useful if we can somehow filter as much information from as little data obtained from the hyperspectral image. Various methods are available in reducing the dimension of a dataset without throwing away much, if any, of the information, and these methods can be applied to the hyperspectral image to give us a reduced hyperspectral image on which we can then perform face recognition techniques at a lower computational cost.

We studied the results of a variety of methods for performing face recognition using the Scale Invariant Transformation Feature (SIFT) algorithm as a matching function on uncorrelated spectral bands, principal component representation of the spectral bands, and the ensemble decision of the two. We conclude that there is no dominating method in the scope of our research; however, we do obtain three methods that outperform the results obtained from a previous study which only considered a SIFT application on a single hyperspectral band, and our method performs very well under temporal variation. Although the data that we used for our research was not as clean as we could have hoped for it to be, it is still beneficial since it better mimics the type of data that would be obtained in the real world versus a very pristine dataset that was obtained under tightly controlled settings.

With the results that we obtained from our research, we can safely suggest that hyperspectral images can provide the additional information required in face recognition methods that allows it to perform well whenever aging or physical changes within the subject’s feature is present. This could be very useful when other biometrics in indentifying the target that would certainly perform well under such condition, such as DNA matching, is not available. Thus, it is likely that we can utilize face recognition the next time we are required to identify a high profile target within the near future when none of the other forms of biometric is available.

Appendix D: Story Board

Face Recognition via Ensemble SIFT Matching of Uncorrelated Hyperspectral Bands and Spectral PCTs

Fairul Mohd-Zaid
Department of Operational Sciences (ENS)

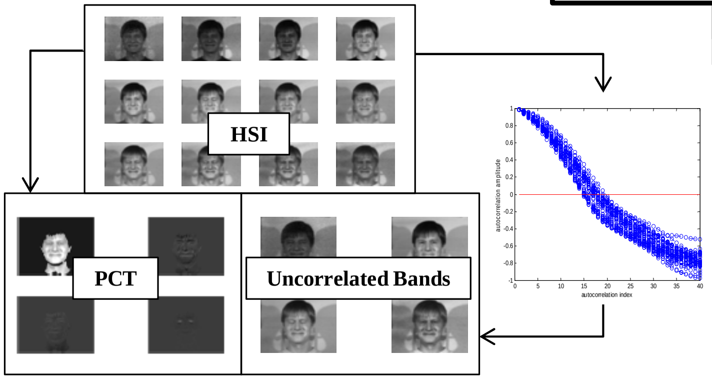
ADVISOR
Dr. Kenneth Bauer

READER
Maj Mark Friend, PhD

Research Objectives:

- Study the performance of applying SIFT on HSI
- Study the performance of ensemble SIFT matching using uncorrelated bands and PCT
- Study the performance of our method with respect to temporal variation

METHODOLOGY



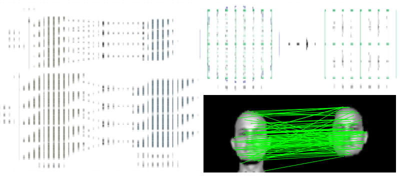
- Hyperspectral face images of 36 subjects are collected between the 450 to 1100 nm spectral range
- Correlation study between spectral bands is performed to obtain uncorrelated bands
- Principle Component Transformations (PCT) are also simultaneously performed on each hyperspectral images to obtain PCT images

Difference of Gaussians

$$D(x,y,\sigma)=L(x,y,k\sigma)-L(x,y,\sigma)$$

$$L(x,y,\sigma)=G(x,y,\sigma)*I(x,y)$$

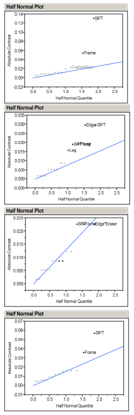
$$G(x,y,\sigma)=\frac{1}{2\pi\sigma^2}e^{\frac{x^2+y^2}{2\sigma^2}}$$



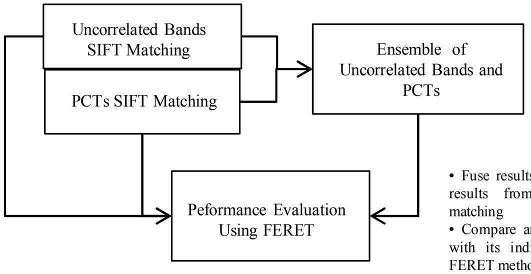
- Apply Scale Invariant Feature Transform (SIFT) on probe and target images to extract keypoints
- Use SIFT keypoints for face recognition
- Perform on both uncorrelated bands and PCT images

	Basic Design					
Run	A	B	C	D	E = ABC	F = BCD
1	-	-	-	-	-	-
2	+	-	-	-	+	-
3	-	+	-	-	+	+
4	+	+	-	-	-	+
5	-	-	+	-	+	+
6	+	-	+	-	-	+
7	-	+	+	-	-	-
8	+	+	+	-	+	-
9	-	-	-	+	-	+
10	+	-	-	+	+	+
11	-	+	-	+	+	-
12	+	+	-	+	-	-
13	-	-	+	+	+	-
14	+	-	+	+	-	-
15	-	+	+	+	-	+
16	+	+	+	+	+	+

Parameter	Low Setting	High Setting
Eraser	5	65
Match	3	7
Lag	16	18
Scales	2	3
Edge	5	10
Frame	2	3



- Use Design of Experiments (DOE) to obtain significant settings
- Obtain optimal settings based on designed experiment



Uncorrelated Bands SIFT Matching

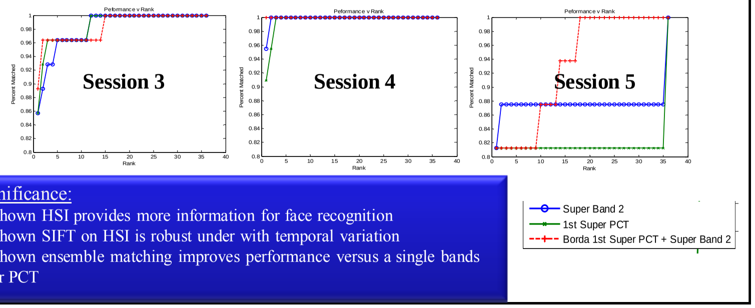
PCTs SIFT Matching

Ensemble of Uncorrelated Bands and PCTs

Peformance Evaluation Using FERET

- Fuse results from uncorrelated bands with results from PCT to obtain ensemble matching
- Compare and contrast ensemble matching with its individual components using the FERET method

RESULTS



Significance:

- Shown HSI provides more information for face recognition
- Shown SIFT on HSI is robust under with temporal variation
- Shown ensemble matching improves performance versus a single bands or PCT

Bibliography

Beveridge, J. R., Bolme, D., Draper, B. A., & Teixeira*, M. (*2005). The CSU Face Identification Evaluation System. *Machine Vision and Applications* , 16: 128–138.

Brown, M., & Lowe, D. *(2002).* Invariant features from interest point groups. *British Machine Vision Conference*, (pp. 656-665). Cardiff, Wales.

Denes, L. J., Metes, P., & Liu, Y. (*2002). Hyperspectral Face Database*. Pittsburgh: Robotics Institute, Carnegie Mellon University.

Eismann, M. T. (2010). *Hyperspectral Remote Sensing OENG 647 Coursenote*. Air Force Institute of Technology.

Elbakary, M. I., Alam, M. S., & Aslan, M. *S. (2007).* Face Recognition Algorithm in Hyperspectral Imagery by Employing the K-means Method and the Mahalanobis Distance. *Proceedings of SPIE* , 6697:1-9.

Healey, G., & Slater, D. (1999). Models and methods for automated material identification in hyperspectral imagery acuired under unknown illuminatino and atmospheric condition. *IEEE Transaction on Geoscience and Remote Sensing* , 2706-2717.

Johnson*,* R. (2008). *Improved Feature Extraction, Feature Selection, and Identification Techniques That Create a Fast Unsupervised Hyperspectral Target Detection Algorithm.* Wright-Patterson AFB: Air Force Institute of Technology.

Kirby, M., & Sirovich, L. (1990). Application of the Karhunen-Loeve procedure for the characterization of human faces. *IEEE Transactions on Pattern Analysis and Machine Intelligence* , 12(1).

Koenderink, J. (1984). The structure of images. *Biological Cybernetics* , 50*: 363-396.*

Landgrebe*, D. A. (2003). Signal Theory Methods in Multispectral Remote Sensing.* Hoboken, New Jersey: John Wiley & Sons, Inc.

Lindeberg*,* T. (1994). Scale-space theory: A basic tool for analysing structures at different scales*. Journal of Applied Statistics ,* 21(2):224-270*.*

Lowe, D. G. (2004). Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision, Vol. 60, No. 2, November 2004* , 91-110.

Lowe, D. G. (1999). Object Recognition from Local Scale-Invariant Features. *Proc. of International Conference on Computer Vision*, (pp. 1150-1157). Corfu, Greece.

Luo, J., Ma, Y., Takikawa, E., Lao, S., Kawade, M., & Lu, B.-L. (2007). Person-specific SIFT features for face recognition. *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007., vol.2,* (15-20 April 2007) , II-593-II-596.

Montgomery, D. C. (2009). *Design and Analysis of Experiments.* New Jersey: John Wiley & Sons, Inc.

Nunez, A. (2009). *A Physical Model of Human Skin and Its Application for Search and Rescue*. Wright-Patterson AFB: Air Force Institute of Technology.

Pan, Z., Healey, G. E., Prasad*,* M., & Tromberg, B. J. (2003). Face Recognition in Hyperspectral Images. *Proceedings., 2003 Computer Society Conference on Computer Vision and Pattern Recognition* , 1-6.

Pan, Z., Healey, G., & Tromberg, B. (2005). Multiband and Spectral Eigenfaces for Face Recognition in Hyperspectral Images. *Proceedings of SPIE Vol. 5779* , 144-151.

Phillips, P. J., Moon, H., Rauss, P., & Rizvi, S. A. (1997). The FERET Evaluation Methodology for Face-Recognition Algorithms. *IEEE* , 137-143.

Polikar, R. (2006). Ensemble Based Systems in Decision Making. *IEEE Circuits and Systems Magazine ,* 21-45.

Ryer*,* D.*,* Bihl, T. J., Bauer, K. W., & Rogers, S. *K. (2011*). QUEST hierarchy for hyperspectral face recognition. *Proceedings of SPIE Vol 8029.*

Shaw, G., & Manolakis, D. (2002). Signal Processing for Hypersectral Image Exploitation*. IEEE Signal Processing Magazine ,* 19, 13.

Sirovich, L., & Kirby, M. (1987). Low-Dimensional procedure for the characterization of human faces. *Journal of the Optical Society of America* , 519-524.

Turk, M., & Pentland, A. (1991). Eigenfaces for Recognition. *Journal of Cognitive Neuroscience Volume 3, Number 1* , 71-86.

Vedaldi, A. (2006). *A. Vedaldi - Code - SIFT for Matlab.* Retrieved December 2010, from A. Vedaldi - Andrea Vedaldi: http://www.vlfeat.org/~vedaldi/code/sift.html

Welch, P. D. (1983). The Statistical Analysis of Simulation Results. In S. S. Lavenberg, *Computer Performance Modeling Handbook*. New York, NY: Academic Press.

Williams, *J. P. (2007). Robustness of Multiple Clustering Algorithms on Hyperspectral Images (Thesis).* Wright-Patterson AFB: Air Force Institute of Technology.

Witkin, *A. P. (1983).* Scale-space filtering*. International Joint Conference on Artificial Intelligence,* (pp. 1019-1022). *Karlsruhe*, Germany.

Xu, Y., Hu, K., Tian, Y., & Peng, F. (2008). Classification of Hyperspectral imagery using SIFT for spectral matching. *Image and Signal Processing, 2008. CISP '08, Congress on Image and Signal Processing, vol.2,* (27-30 May 2008) , 704-708.

Zhao, W., Chellappa, R., J., P. P., & Rosenfeld, A. (2003). Face Recognition: A literature survey. *ACM Computing Surveys* , 399-458.

Vita

Mr. Fairul Mohd-Zaid attended the Southern Polytechnic State University in Marietta, Georgia, upon graduating from Norcross High School in Norcross, Georgia, in May 2004. He then graduated as a Bachelor of Science in Mathematics in May 2008 and spent a year as an intern to the Chief Scientist of the Air Mobility Command (AMC/ST) at Scott AFB.

There he assisted the Chief Scientist in evaluating the possibilities of adopting new technologies into the operations of AMC and he also coordinated numerous ST hosted visits by other Department of Defense (DoD) agencies and industries; including visits by General Officers and members of the Senior Executive Services. He was also the Executive Officer of AMC's Science and Technology Working Group (STWG) chaired by the Chief Scientist and was involved in the Formation Flying For Fuel Efficiency Red/Blue Study which was presented to the Commander of AMC.

He then entered the Air Force Institute of Technology's Operations Research Master's program in September 2009 under the DoD funded SMART Scholarship where he will continue conducting research at the 711th Human Performance Wing Human Effectiveness Directorate, 711HPW/RH, at Wright-Patterson AFB upon graduation.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

1. REPORT DATE *(DD-MM-YYYY)* 16-06-2011	**2. REPORT TYPE** **Master's Thesis**	**3. DATES COVERED** *(From – To)* Sep 2010 - Jun 2011

4. TITLE AND SUBTITLE Face Recognition via Ensemble SIFT Matching of Uncorrelated Hyperspectral Bands and Spectral PCTs	**5a. CONTRACT NUMBER** **5b. GRANT NUMBER** **5c. PROGRAM ELEMENT NUMBER**
6. AUTHOR(S) Mohd-Zaid, Mohd Fairul	**5d. PROJECT NUMBER** **5e. TASK NUMBER** **5f. WORK UNIT NUMBER**
7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S) Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 Hobson Street, Building 642 WPAFB OH 45433-7765	**8. PERFORMING ORGANIZATION REPORT NUMBER** AFIT-OR-MS-ENS-11-16
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) AFRL/RY 2241 Avionics Circle Area B Bldg 620 WPAFB OH 45433-7765	**10. SPONSOR/MONITOR'S ACRONYM(S)** NASIC/DAIA AFRL/RYZT **11. SPONSOR/MONITOR'S REPORT NUMBER(S)**

12. DISTRIBUTION/AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Face recognition is not a new area of study, but facial recognition using through hyperspectral images is a somewhat new concept which is still in its infancy. Although the conventional method of face recognition using Red-Green-Blue (RGB) or grayscale images has been advanced over the last twenty years, these methods are still shown to have weak performance whenever there are variations or changes in lighting, pose, or temporal aspect of the subjects. A hyperspectral representation of an image captures more information that is available within a scene than a RGB image therefore it is beneficial to study the performance of face recognition using a hyperspectral representation of the subjects' faces. We studied the results of a variety of methods for performing face recognition using the Scale Invariant Transformation Feature (SIFT) algorithm as a matching function on uncorrelated spectral bands, principal component representation of the spectral bands, and the ensemble decision of the two. We conclude that there is no dominating method in the scope of our research; however, we do obtain three methods with leading performances despite some trade-off between performance at lower ranks and performance at higher ranks…that outperform the results obtained from a previous study which only considered a SIFT application on a single hyperspectral band which also performs very well under temporal variation.

15. SUBJECT TERMS
Face Recognition, SIFT, Ensemble Classification, Fusion, Diversity, PCA, Hyperspectral Imaging (HSI), Design of Experiment

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** Dr. Kenneth W. Bauer (ENS)
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			**19b. TELEPHONE NUMBER** *(Include area code)* (937) 255-3636, ext 4631; e-mail: Kenneth.Bauer@afit.edu
U	U	U	UU	107	